UNIVERSIDADE FEDERAL DO PARÁ
CENTRO DE GEOCIÊNCIAS
CURSO DE PÓS-GRADUAÇÃO EM GEOFÍSICA

DISSERTAÇÃO DE MESTRADO

REESTRUTURAÇÃO E AMPLIAÇÃO DA eGs (electrical GEOPHYSICS suite)

GLAUCO LIRA PEREIRA

BELÉM-PARÁ

2005

UNIVERSIDADE FEDERAL DO PARÁ

CENTRO DE GEOCIÊNCIAS

CURSO DE PÓS-GRADUAÇÃO EM GEOFÍSICA

REESTRUTURAÇÃO E AMPLIAÇÃO DA eGs (electrical GEOPHYSICS suite)

DISSERTAÇÃO APRESENTADA POR

GLAUCO LIRA PEREIRA

COMO REQUISITO PARCIAL À OBTENÇÃO DE GRAU DE MESTRE EM CIÊNCIAS NA ÁREA DE GEOFÍSICA

Data de Aprovação : 14/03/2005

Nota: 90% (EXCELENTE)

BANCA EXAMINADORA:

Dr. Luiz Rijo (Orientador)

Dr. José Gouvêa Luiz

Dr. Paulo Roberto de Carvalho

BELÉM

2005

Aos meus pais pelo apoio completo,
as minhas sobrinhas e sobrinho pelos carinhos,
e a Carla Lopes pela paciência.

AGRADECIMENTOS

Gostaria de expressar meus agradecimentos ao Prof. Luiz Rijo por sua competente orientação e apoio dispensados a mim durante o desenvolvimento desta dissertação e pela diversão proporcionada pelo projeto eGs.

Aos membros da Banca Examinadora, pelas críticas construtivas e importantes sugestões.

Aos colegas contemporâneos de curso Marcos Welby, Victor Tocantins, Frayzer Almeida, Gilberto Vogado e Rodrigo Erasmo, pelo apoio, incentivo e momentos de descontração.

Ao corpo docente do CPGf/UFPa, que não mede esforços para manter um curso de pós-graduação de bom nível.

A secretária Benildes Lopes (CPGf/UFPa), sempre disposta a ajudar nas questões administrativas.

A CNPq, pela concessão de bolsa de estudo, sem a qual a conclusão deste trabalho seria mais difícil, e ao PRH/ANP-06 por auxiliar na infra-estrutura computacional e física do PROEM (Laboratório de Processamento Eletromagnético do CG/UFPa).

Finalmente gostaria de agradecer à minha família, em especial ao meu irmão Glautier, por estar sempre disposto a ajudar.

SUMÁRIO

DEDICATÓRIA i

AGRADECIMENTOS ii

LISTA DE FIGURAS v

RESUMO 1

ABSTRACT 2

1 INTRODUÇÃO 3

2 MOTIVAÇÂO E ESTRATÉGIA. 7
2.1 - MOTIVAÇÂO. 7
2.2 - ESTRATÉGIA. 8
2.2.1 - **Concepção e Elaboração** 8
2.2.2 - **Princípio da Decomposição** 11
2.2.3 - **Princípio do Refinamento** 12
2.2.4 - **Junção das Soluções** 14

3 A EGS 3.0 E SUA APLICABILIDADE 15
3.1 - CONHECENDO A eGs 15
3.1.1 - **Distribuição** 16
3.1.2 - **Instalação** 17
3.1.3 - **Inicialização do Programa** 17
3.1.4 - **Janela "eGs 3.0"** 18
3.1.5 - **Janela "Métodos"** 23
3.1.6 - **Janela "Seleção"** 24
3.1.7 - **Janelas "Modelo"** 26
3.1.8 - **Janelas "Dados Observados"** 27
3.2 - EXEMPLOS 29
3.2.1 - **1D - Uma dimensão** 30
3.2.2 - **2D - Duas dimensões** 31

4 DESENVOLVIMENTO 40
4.1 - CONSTRUÇÃO. 40
4.1.1 - **Janela "eGs 3.0"** 40
4.1.2 - **Janela "Métodos"** 45
4.1.3 - **Janela "Seleção"** 46
4.1.4 - **Janelas "Modelo"** 50
4.1.5 - **Janelas "Dados Observados"** 54
4.2 - O USO DAS DLLS 68

5 CONCLUSÕES E SUGESTÕES 74

REFERÊNCIAS BIBLIOGRÁFICAS 76

APÊNDICE A 78

LISTA DE FIGURAS

2.1 Diagrama de entrada e saída de dados 9
2.2 Fluxograma do pseudocódigo 10
2.3 Diagrama de pré-processamento dos dados 11
2.4 Diagrama de processamento dos dados 11
2.5 Diagrama de pós-processamento dos dados 12

3.1 Ícone da eGs 3.0 17
3.2 Tela de apresentação da eGs 3.0 17
3.3 Aplicativo eGs 3.0 18
3.4 Janela eGs 3.0 19
3.5 Menu da eGs 3.0 com a barra de ferramentas fora da janela principal 19
3.6 Janela modal para salvar arquivos 23
3.7 Árvore de Métodos da eGs 3.0 24
3.8 Janela Seleção da eGs 3.0 25
3.9 (a)Janela que informa para selecionar um método (b)Janela que informa para selecionar o tipo de problema (direto ou inverso). 25
3.10 (a) Janela padrão do *modelo geoelétrico* 1D. (b) Janela "Modelo" com a opção *permeabilidade* selecionada para alterar o valor. 26
3.11 Janela informativa para o número de camadas. 27
3.12 Janela informativa ao editar a última espessura. 27
3.13 (a)Janela para os *dados observados* da SEV (b)Janela para os *dados observados* da sondagem magnetotelúrica para simular o eletrojato equatorial. 28
3.14 Janela para os *dados observados* de IP e Resistividade. 30
3.15 (a)*modelo geoelétrico* com 5 camadas assim distribuídas: 40 Ω-m (350 m), 150 Ω-m (1000 m), 10 Ω-m (4000 m), 500 Ω-m (42500 m) e 10 Ω-m; (b)*dados observados* em 26 freqüencias. 31
3.16 (a) Curva da resistividade da resposta do *modelo geoelétrico* "MgeMT1d" com 0% de ruído e dos *dados observados* "ObsMT1d". (b) Curva da fase da impedância da resposta do *modelo geoelétrico* "MgeMT1d" com 0% de ruído e dos *dados observados* "ObsMT1d". (c) Curva da resistividade da resposta do *modelo geoelétrico* "MgeMT1d" com 3% de ruído e dos *dados observados* "ObsMT1d". (d) Curva da fase da impedância da resposta do *modelo geoelétrico* "MgeMT1d" com 3% de ruído e dos *dados observados* "ObsMT1d". 32

3.17 Malha do *modelo geoelétrico* "resIp2D181x18". 33
3.18 (a)Visualização da pseudo-seção da resistividade aparente editada, (b) Interpolação Otimizada da pseudo-seção da resistividade aparente editada. 34
3.19 (a)Visualização da pseudo-seção do PFE editado, (b) Interpolação Otimizada da pseudo-seção do PFE editado. 35
3.20 (a)Visualização da pseudo-seção do fator metálico editado, (b) Interpolação Otimizada da pseudo-seção do fator metálico editado. 36
3.21 (a)Visualização do resultado da modelagem numérica da pseudo-seção da resistividade aparente, (b) Interpolação Otimizada do resultado da modelagem numérica da pseudo-seção da resistividade aparente. 37
3.22 (a)Visualização do resultado da modelagem numérica da pseudo-seção do PFE, (b) Interpolação Otimizada do resultado da modelagem numérica da pseudo-seção do PFE. 38
3.23 (a)Visualização do resultado da modelagem numérica da pseudo-seção do fator metálico, (b) Interpolação Otimizada da pseudo-seção do resultado da modelagem numérica do fator metálico. 39

4.1 Menu *pulldown* da eGs 3.0 41
4.2 (a) *modelos geoelétricos* de um meio estratificado (b) Formato do arquivo para uma dimensão. 51
4.3 Formato do arquivo para uma dimensão. 55
4.4 (a) Representação geométrica do eletrojato equatorial 3 gaussiana (b) Formato do arquivo para eletrojato equatorial 3 gaussiana. 56

A.1 Ícone de instalação da eGs 3.0 79
A.2 Janela de boas-vindas para instalação do programa 79
A.3 Informação para instalação 79
A.4 Termo de licença 80
A.5 Informações do diretório, do espaço necessário e livre no disco rígido 80
A.6 Janela de confirmação da instalação 81
A.7 Janela que informa a finalização da instalação do programa 81
A.8 Janela Métodos 82
A.9 Janela "Seleção" para abrir um *modelo geoelétrico* novo 83
A.10 Abrir um *modelo geoelétrico* novo através do Menu 84
A.11 Janela "Modelo" para início da edição de um *modelo geoelétrico* 84

A.12 *Modelo geoelétrico* 1D usado como exemplo 85
A.13 (a) Janela "Modelo" com permeabilidade relativa igual a 1 em todas camadas.(b) Janela "Modelo" para alteração da permeabilidade magnética relativa 86
A.14 Opção Salvar Modelo do Menu 87
A.15 Janela de aviso quando um *modelo geoelétrico* está sendo fechado, mas as alterações não foram salvas 87
A.16 Janela "Seleção" para abrir um *modelo geoelétrico* salvo 88
A.17 Abrir um *modelo geoelétrico* salvo através do Menu 88
A.18 Janela amigável para abertura de arquivos 89
A.19 Janela "Modelo" aberta para reedição de um arquivo salvo anteriormente .. 89
A.20 Janela "Seleção" para abrir a janela "Dados Observados" nova 90
A.21 Abrir uma janela "Dados Observados" nova através do Menu 91
A.22 Nova janela "Dados Observados" para início da edição 91
A.23 Nova janela "Dados Observados" editada 92
A.24 Opção Salvar Dados Observados do Menu 93
A.25 Janela de aviso quando os dados observados estão sendo fechados, mas as alterações não foram salvas 93
A.26 Janela "Seleção" para abrir dados observados salvos 94
A.27 Abrir os dados observados salvos através do Menu 94
A.28 Janela amigável para abertura de arquivos 95
A.29 Janela "Dados Observados" aberta para reedição de um arquivo salvo anteriormente 95
A.30 (a) Janela "Seleção" com o botão Executar ativado (b) Itens Executar Com Gráfico e Executar Sem Gráfico 96
A.31 Resultado da modelagem numérica com o item Executar Com Gráfico ativado 96
A.32 Ilustração de uma modelagem usando o arquivo Mt1D.MGE como *modelo geoelétrico* 97
A.33 Item Abrir Arquivo 98
A.34 Sub-menus do menu Gráfico 99
A.35 Janela que permite escolher as escalas de cada gráfico 100
A.36 Janela que permite escolher quais colunas do arquivo serão usadas para gerar o gráfico 100
A.37 Janela que permite escolher os intervalos de visualização dos gráficos 101

A.38 Janela que permite escolher os textos que identificam a janela, os eixos e as curvas 102
A.39 Janela que permite escolher os símbolos que representarão os pontos nos gráficos 102
A.40 Janela Métodos 104
A.41 Janela "Seleção" para abrir um *modelo geoelétrico* novo 105
A.42 Abrir um *modelo geoelétrico* novo através do Menu 106
A.43 Janela "Seleção" para abrir um *modelo geoelétrico* salvo 106
A.44 Abrir um *modelo geoelétrico* salvo através do Menu 107
A.45 Janela usada para editar o modelo 2D 107
A.46 Barra de ferramentas da janela usada para desenhar o *modelo geoelétrico* ... 108
A.47 Janela usada para definir as dimensões da malha usada no *modelo geoelétrico* do método IP e Resistividade 2D 110
A.48 Ilustração da forma do *modelo geoelétrico* usado na modelagem usando o método IP and Resistividade 2D 111
A.49 Campos destinados à declaração das propriedades físicas do *modelo geoelétrico* 111
A.50 Opção Salvar Modelo do Menu 112
A.51 Janela de aviso quando um *modelo geoelétrico* está sendo fechado, mas as alterações não foram salvas 112
A.52 Janela "Seleção" para abrir a janela "Dados Observados" nova 113
A.53 Abrir uma janela "Dados Observados" nova através do Menu 113
A.54 Nova janela "Dados Observados" para início da edição 114
A.55 Nova janela "Dados Observados" editada 114
A.56 Escolha para edição da resistividade aparente, percentual do efeito da freqüência e fator metálico ou apenas da resistividade aparente 115
A.57 Escolha do tipo de propriedade o usuário deseja editar 115
A.58 Opção Salvar Dados Observados do Menu 116
A.59 Janela de aviso quando os dados observados estão sendo fechados, mas as alterações não foram salvas 116
A.60 Janela "Seleção" para abrir dados observados salvos 117
A.61 Abrir os dados observados salvos através do Menu 117
A.62 Janela para abertura de arquivos 118
A.63 Janela "Dados Observados" aberta para reedição de um arquivo salvo ante-

riormente 118
A.64 (a) Janela “Seleção” com o botão Executar ativado. (b) Itens *Executar Com Gráfico* e *Executar Sem Gráfico* 119
A.65 Resultado da modelagem numérica com o item *Executar Com Gráfico* ativado 120
A.66 Resposta do *modelo geoelétrico* do método IP e resistividade 121
A.67 Opção para interpolação das pseudo-seções de Resistividade e IP (deve estar marcada a opção "sim") 122

RESUMO

O presente trabalho faz parte de um projeto, coordenado pelo Prof. Luiz Rijo, para o desenvolvimento de uma suíte de programas denominada de *electrical Geophysics suite* (eGs) destinada a alunos de graduação e pós-graduação em geologia e geofísica para modelagem e interpretação de dados geofísicos elétricos e eletromagnéticos. Esta dissertação consiste da descrição na reestruturação e atualização (ampliação) do módulo de pré-processamento (entradas de dados) da terceira versão da referida suíte. A eGs é um conjunto de interfaces gráficas desenvolvida em Delphi para uso de bibliotecas de ligação dinâmica (dll - *Dynamic Link Libraries*) feitas em Fortran para modelagem numérica e interpretação de dados geofísicos elétricos e eletromagnéticos. O principal objetivo foi estruturar a arquitetura da suíte para que o usuário não sentisse dificuldades em usá-la em aplicações geofísicas. Os resultados apresentados consistem no uso do *software* em aplicações específicas de simulação e exploração, mostrando que o objetivo foi alcançado com sucesso. A eGs faz parte do paradigma de *softwares* livres (distribuídos gratuitamente) que tenha alto grau de confiabilidade e rapidez em armazenar, processar e interpretar os dados coletados em campanhas de campo, sem ter de assimilar alguma linguagem de programação. O desenvolvimento de um software como a eGs é um empreendimento contínuo que deve ser levado a frente com futuras versões. A participação dos usuários, com sugestões e comentários de erros (bugs) no programa, é importante para o aprimoramento da suíte nas novas versões. A mim coube reestruturar, atualizar e desenvolver a estrutura de entradas de dados de modelos unidimensionais e adaptar a suíte para a incorporação de modelos 2D e 3D. Um manual de usuário, na forma de um Apêndice, é apresentado àqueles que desejam usar, de fato, a suíte. Aos os que desejam dar continuidade ao projeto da eGs são apresentados muitos detalhes técnicos do código fonte do software. Estes detalhes são para especialistas e são totalmente transparentes ao usuário final da suíte. A eGs é de distribuição livre (grátis) e pode ser obtida em *www.rijo.pro.br*.

ABSTRACT

The present work is part of a project coordinated by Prof. Luiz Rijo for the development of a suite of programs called electrical Geophysics suite (eGs), to be used to geology and geophysics undergraduate and graduated students, for modelling and interpretation of electric and electromagnetic geophysical data. This dissertation describes of the restructuring and updating (enlargement) of the preprocessing module (input of data) of the third version of the referred suite. The eGs is a group of graphic interfaces developed in Delphi for use of libraries of dynamic link (dll - Dynamic Link Libraries) developed in Fortran for numeric modelling and interpretation of electric and electromagnetic geophysical data. The main objective of this work was to structure the architecture of the suite so that the user does not feel difficulties in using it in geophysical applications. The accompanied results consist of the software capability in specific applications for simulation and exploration, showing that the objective was reached with success. The eGs is part of the paradigm of free software (distributed at no cost) that has high degree of reliability and speed in storing, processing, and interpreting geophysical data collected in field campaigns, without having to assimilate any computer programming language. The development of a software as the eGs is a continuous undertaking that should be continued with new versions. The users' participation with suggestions and comments on bugs in the program is important for the improvement of the suite in future versions. My participation was to restructure, to update, and to develop the structure of graphic interfaces for input of data for 1D modelling and to prepare the suite for the incorporation of interfaces for 2D and 3D modelling. An User Manual, in the form of an Appendix, is presented for those that want use effectively the suite. Many details of the source code of the software were included for those that want to continue the project of the eGs. These details are for specialists and they are totally transparent to the final user of the suite. The eGs is a free software and it can be obtained in *www.rijo.pro.br*.

1 - INTRODUÇÃO

Os métodos elétricos e eletromagnéticos têm a finalidade de determinar as propriedades elétricas e geométricas dos meios geológicos da subsuperfície da terra. Para isso, são utilizadas técnicas que empregam correntes elétricas ou ondas eletromagnéticas geradas através de fontes de campo elétrico ou magnético naturais ou artificiais. Com isso, obtêm-se informações das características das rochas de forma indireta, ou seja, sem a necessidade de poço para observar e estudar a geologia na subsuperfície do terreno.

Os métodos elétricos e eletromagnéticos são empregados em estudos geotectônicos, na exploração de petróleo, na prospecção mineral, assim como em estudos ambientais de águas subterrâneas. Dependendo da escala, em cada um desses casos, há várias técnicas de exploração elétrica e eletromagnética. Por exemplo, o método magnetotelúrico é aplicado em estudos geotectônicos, o método MCSEM (*Marine Controlled Source ElectroMagnetic*) e perfilagem de indução são usados na exploração de petróleo, os métodos eletromagnéticos clássicos, como Slingram e o elétrico polarização induzida (IP) são utilizados na prospecção mineral, sondagens elétricas verticais são empregadas na prospecção de água subterrânea e o radar de penetração no solo (GPR) em estudos ambientais.

A teoria de todos esses métodos é baseada nas equações de Maxwell. Estas equações podem ser avaliadas por métodos analíticos, analógicos e numéricos. Os métodos analíticos se restringem a modelos mais simples, meios estratificados e, conseqüentemente, altamente limitados para aplicações mais realistas. O procedimento analógico é usado para geometrias um pouco mais complexas e com condutividade variável, sendo mais comumente usada para interpretações eletromagnéticas na prospecção mineral. Essa técnica de modelagem faz uso da identificação por escala, baseada na teoria de similitude, sendo muito versátil na simulação em laboratório de modelos geológicos reais em escalas reduzidas. A modelagem numérica, por sua vez, é utilizada para modelos em duas e três dimensões muito mais complexos do que as soluções analíticas. Quase toda distribuição arbitrária da condutividade pode ser modelada usando a aproximação numérica por diferenças finitas, equações integrais ou elementos finitos. O principal objetivo do método numérico é desenvolver algoritmos eficientes que possam ser utilizados em microcomputadores ou *clusters* para interpretação de *dados observados* no campo. Pensando nisso, o Prof. Luiz Rijo teve a idéia de criar um conjunto de programas para servir aos métodos geofísicos elétricos e eletromagnéticos o qual resultou na chamada eGs (electrical Geophysics suite).

O projeto eGs é um conjunto de programas para modelagem direta e modelagem inversa de dados de levantamentos geofísicos pelos métodos elétricos e eletromagnéticos. A suite vem sendo desenvolvida há vários anos pelo Prof. Luiz Rijo, inicialmente denominada de EGSLIB (*Electrical Geophysics Software LIBrary*). No início da década de setenta, a EGSLIB consistia em uma biblioteca de programas desenvolvidos em FORTRAN 77 e executados em *mainframes* que utilizavam perfuradoras e leitoras de cartões Hollerith.

Posteriormente, com o começo da evolução da informática, os programas começaram a ser utilizados em uma nova categoria de plataforma, estações de trabalho RISC/UNIX. Nesse novo formato, o uso dos programas da EGSLIB era feito pelos usuários através de comandos, às vezes bastante difíceis, na tela de um monitor. Desse modo, o caminho para modelagem dos dados até a interpretação continuava sendo trabalhoso, pois era necessária uma estação de trabalho dispendiosa e complexa.

A partir do surgimento dos computadores pessoais, a EGSLIB passou a ser utilizada junto a outros programas como MATLAB, que tem uma grande variedade de opções para criação de gráficos complicados. Porém, o usuário deveria compreender bem a linguagem de programação do MATLAB e fazer a formatação do texto nos arquivos de dados de entrada para cada tipo de método geofísico elétrico ou eletromagnético.

Devido aos dados nos arquivos de entrada serem editados em um arquivo de texto, tornando-se, em alguns casos, o processo mais demorado para obtenção dos dados modelados, pois os valores eram digitados um a um. Então, para facilitar isto, em 1998, surgiu um esboço, para o ambiente operacional *Windows*, criado por Rodrigo de Freitas Valle, aluno do curso de informática da UFPA e bolsista de iniciação cientifica sob a responsabilidade do Prof. Luiz Rijo, que objetivava desenvolver uma interface gráfica para facilitar a edição e visualização dos arquivos de entrada de dados, porém os pacotes dos programas em FORTRAN 77 ainda eram executados em DOS, fora do ambiente operacional *Windows*. A visualização dos resultados era feita através de outros programas, por exemplo, EASY PLOT e MATLAB.

Posteriormente, para fazer com que a manipulação da EGSLIB pelo usuário fosse direta, ou seja, sem auxilio de nenhum outro programa, Monteiro (2001) desenvolveu a arquitetura da interface gráfica para a escolha dos métodos elétricos e eletromagnéticos e a mudança operacional dos programas da EGSLIB, que passaram de programas FORTRAN, agora na versão 90, para bibliotecas de ligação dinâmica, as dlls (*Dynamic Link Libraries*), assim a EGSLIB deu origem à eGs versão 2000. Com isso, a tela do DOS desapareceu por completo.

Após a construção da primeira versão (EGSLIB) e da segunda (eGs 2000), a eGs chega

a sua terceira versão, denominada de eGs 3.0, a qual foi desenvolvida concomitantemente com o aluno Rodrigo E. da Conceição Silva (Silva, R. E. C., 2005), entretanto com o mesmo objetivo das versões anteriores, isto é, um *software* de distribuição livre baseado em um sistema operacional bastante difundido (MS - Windows). Com uma interface gráfica de fácil uso para a escolha do método geofísico elétrico e eletromagnético, edição de dados, criação dos arquivos para modelagem e gráficos para interpretação, sem ter a necessidade de assimilar alguma linguagem de programação.

As mudanças mais significativas entre as versões anteriores (EGSLIB e eGs 2000) e a atual (eGs 3.0) estão na aparência, na arquitetura e na atualização de várias dlls em FORTRAN 90 para FORTRAN 95. Na estética do programa foram alterados o ícone, a tela de apresentação, o menu, a janela que escolhe o método e as janelas para modelo e dados observados. A estrutura da eGs foi repensada para facilitar a interatividade do usuário com o programa. Ocorreram, portanto, ampliações nas opções para os gráficos e às malhas de elementos finitos 2D e 3D, mudanças na maneira de editar e acessar os arquivos e, conseqüentemente, foi acrescentado um formulário chamado de seleção e o arquivo de ajuda. Em outras palavras, a terceira versão é praticamente um novo programa quando comparado às duas primeiras versões.

Para facilitar a apresentação deste trabalho é necessário esclarecer alguns pequenos detalhes técnicos de programação. O primeiro é o conceito do código fonte, que é um conjunto de instruções codificadas em uma determinada linguagem de programação em um ambiente de desenvolvimento com um compilador para a criação do *software* aplicativo. O segundo é o termo genérico *software* aplicativo (também chamado, separadamente, de *software* ou aplicativo), que é utilizado para nomear todas as espécies de programas, ou seja, o resultado da compilação. São os fornecedores de informações aos computadores para desempenhar uma determinada tarefa, podendo nela existir a interação humana. No caso presente, o código fonte da interface gráfica foi escrito no ambiente de programação Delphi com a linguagem Object Pascal e as dlls foram codificadas em linguagem FORTRAN 95.

Daqui para frente, sempre que for citada a palavra formulário estará se fazendo referência à componente do Delphi com um código fonte, chamado de "*unit*", para criação de janelas em um *software*, ou seja, no momento do desenvolvimento do programa, antes da compilação. Por outro lado, quando for citado *software*, programa, ou aplicativo estaremos nos referindo ao *software* aplicativo que possui janelas e sub-janelas, onde ocorre a interação computador-usuário. Cada janela será citada com o seu título entre aspas.

Esta dissertação é constituída de 5 capítulos. O capítulo 1 visa dar uma breve introdução histórica da eGs, explicar alguns conceitos básicos de programação e informar a estrutura da dissertação. O capítulo 2 é formado por duas seções, sendo que a primeira seção tem a finalidade de mostrar os motivos que levaram a reestruturação e atualização da eGs; e a segunda seção tem o intuito de descrever a estratégia usada para cercar o problema e como solucioná-lo. O capítulo 3 é voltado ao público alvo do programa, onde é mostrada a intenção do aplicativo, que é facilitar a modelagem numérica, com um exemplo ilustrativo para 1D e outro para 2D, ressaltando que os modelos 3D são apresentados em outra dissertação (Silva, R. E. C., 2005). O capítulo 4 é destinado aos desenvolvedores de *software* interessados na manutenção e ampliação da eGs, onde a primeira seção mostra a construção da interface gráfica de edição de dados dos métodos, ou seja, expõe alguns procedimentos importantes; e a segunda seção explica como é feita a ligação dos dados de entrada (pré-processamento) com as dlls (processamento). O capítulo 5 contém as conclusões e ressalta a importância deste trabalho para a comunidade geofísica e dá sugestões de aprimoramento às futuras versões da eGs. E, por fim, no final da dissertação existe um Apêndice, o Manual da eGs, que explica aos usuários como utilizar o aplicativo.

2 - MOTIVAÇÂO E ESTRATÉGIA.

2.1 - MOTIVAÇÂO.

Os métodos elétricos e eletromagnéticos devido a facilidade de sua aplicação, boa resposta e baixo custo, são amplamente utilizados na prospecção mineral, na exploração de água subterrânea, em estudos ambientais e na indústria de petróleo e, que envolvem uma grande variedade de ambientes geológicos, pois para cada recurso natural que se deseja explorar, existe uma característica específica de alocação do objeto de estudo. Além desses, os métodos elétricos e eletromagnéticos são também aplicados em estudos geotectônicos, dirigidos geralmente, a processos geolégicos que ocorrem em profundidades muito maiores que os acima citados.

Portanto, em termos de geologia, à obtenção de resultados confiáveis, é necessária a análise e interpretação de dados geofísicos elétricos e eletromagnéticos, o que nem sempre é uma tarefa trivial, principalmente nas regiões tropicais, em especial quando o objeto de estudo está na Amazônia, pois o caráter condutivo das rochas e a presença de uma camada superficial intemperizada agem como uma blindagem eletromagnética, interferindo na coleta dos dados observados e nos resultados obtidos.

Tendo orientado 25 dissertações de mestrado e 5 teses de doutorado (veja a referências bibliográficas e *www.rijo.pro.br*), o Prof. Luiz Rijo possui muitos resultados de pesquisas para o desenvolvimento de vários algoritmos baseados em métodos numéricos avançados (elementos finitos, equação integral, métodos dos momentos) implementados na forma de programas de computadores, os quais têm sido amplamente usados por pesquisadores, prospectores e estudantes.

O motivo inicial que estimulou a fazer uma interface gráfica foi o fato de haver, dentro do mercado geofísico, poucos aplicativos de baixo custo, principalmente para estudantes. Os excelentes programas, tanto na aplicabilidade quanto na facilidade de uso, possuem um valor altíssimo, praticamente proibitivos a professores e alunos. Uma solução seria utilizar parte dos executáveis feitos por alunos do curso de pós-graduação em Geofísica da UFPA, porém com a necessária compreensão dos seus códigos fonte, surgindo assim três problemas: os algoritmos e os programas desenvolvidos pelos alunos nem sempre são bem estruturados, não são bem documentados e, em geral, são perdidos quando o estudante conclui o curso. Desse modo, para evitar que o mesmo algoritmo seja reescrito várias vezes, foi criada a eGs.

Porém, as primeiras versões do projeto tiveram que ser atualizadas e ampliadas, pois a necessidade atual é fazer uma interface gráfica auto-sustentável e amigável para qualquer tipo de usuário, desde o iniciante até o mais avançado, de modo que a pessoa, ao optar por utilizar a eGs, possa sentir segurança para fazer todos os processos computacionais para os estudos geofísicos dos métodos implementados no programa. A eGs é constituída de três módulos. O primeiro módulo é denominado de pré-processamento e corresponde a edição de entrada de dados pelo usuário. O segundo módulo (transparente ao usuário), chamado de processamento, é o engenho que executa todas as operações matemáticas dos algoritmos numéricos de modelagem. O terceiro módulo, chamado de pós-processamento, que é constituído por recursos gráficos para apresentação e visualização dos resultados da modelagem.

2.2 - ESTRATÉGIA.

A estratégia utilizada para dar inicio ao desenvolvimento deste trabalho foi reunir os motivos que levaram à elaboração do projeto eGs e verificar quais as reais necessidades para sua atualização. Sendo assim, seguiram-se os seguintes passos:

1° Passo: Concepção e Elaboração;

2° Passo: Princípio da decomposição;

3° Passo: Princípio do refinamento;

4° Passo: Junção das soluções.

2.2.1 - Concepção e Elaboração

Para iniciar o planejamento de um programa, é de extrema importância que o programador saiba se orientar na edição do código-fonte do programa. Para isso, deve-se ter basicamente os seguintes questionamentos:

- Qual é o propósito e quais são os objetivos do programa? (o que ele deve realizar);
- Que linguagem de programação será utilizada?
- Quais os dados necessários para a compilação do programa?
- Qual deve ser o resultado após a compilação?

De posse daquilo que o programa deve realizar, é fundamental listar os processos necessários para transformar os arquivos úteis à compilação em saídas desejadas, e por fim codificar o programa na linguagem escolhida.

A fim de responder à primeira questão, foi proposta uma versão atualizada que fosse mais acessível a um público alvo iniciante em ambiente *Windows*. Pensou-se em uma pessoa que esteja se adaptando tanto ao programa quanto ao padrão da plataforma da *Microsoft*.

Depois de definido o usuário padrão, foi preciso determinar os processos de manipulação dele na eGs, de maneira que o caminho da edição à interpretação dos dados seja menos complicado. Portanto, antes de codificar o programa, identificaram-se interdependências e ocorreram eliminações de entradas redundantes, surgindo o diagrama mostrado na Figura 2.1.

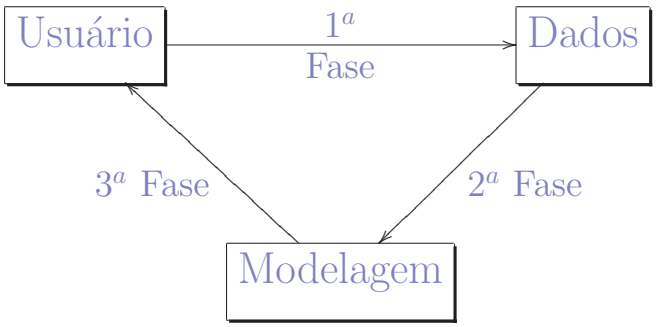


Figura 2.1: Diagrama de entrada e saída de dados

Sendo que a primeira fase é de entrada dos dados do *modelo geoelétrico* interpretativo e dos *dados observados* no campo, ou seja, as observações. A segunda fase, é o armazenamento das informações e processamento. A terceira fase, é a visualização do resultado final da modelagem numérica.

Como o sistema operacional considerado para a utilização da eGs foi o *Windows*, escolheu-se criar dlls em FORTRAN 95, que é um engenho de computação cientifica intensa, fazendo com que os cálculos matemáticos, que exigem muito da máquina dêem os resultados em pouco tempo.

O próximo passo foi a escolha da linguagem para desenvolvimento do *software* e o ambiente de desenvolvimento. Assim, a tecnologia que melhor se adequou às necessidades para a criação dos recursos gráficos (janelas) de entrada e saída de dados da eGs foi o Delphi. Esse ambiente de programação foi escolhido porque as versões anteriores foram desenvolvidas nele, a minha experiência em programar no Delphi e por ser uma ADR (Aplicação de Desenvolvimento Rápido) de desenvolvimento visual, que permite a construção de aplicações

tanto gráficas como em modo texto e que utiliza a linguagem Pascal orientada a objetos, a qual se baseia nos três conceitos fundamentais da programação orientada a objetos (POO): classe, herança e polimorfismo.

Para efeito de elaboração do projeto eGs, foi preciso criar um pseudocódigo com o caráter de auxiliar na edição do algoritmo. Um pseudocódigo é uma narrativa das instruções a serem realizadas pelo programa que pode ou não ser escrito na linguagem de programação. A função do falso código é fazer um roteiro do que será construído para depois implementá-lo na linguagem escolhida, de modo que o resultado após a compilação, sem erros de programação, seja o esperado. A Figura 2.2 mostra o pseudocódigo em forma de fluxograma.

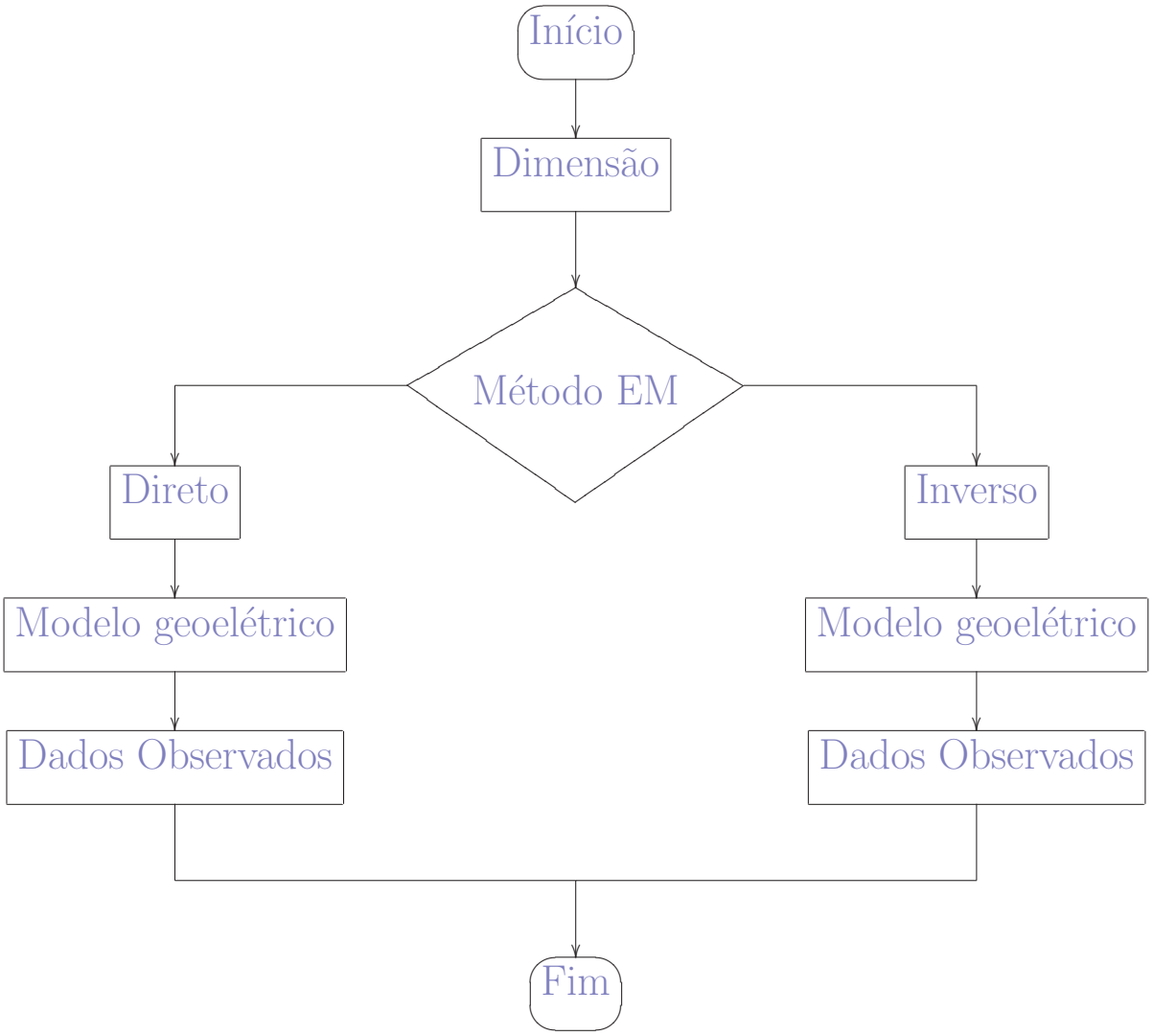


Figura 2.2: Fluxograma do pseudocódigo

2.2.2 - Princípio da Decomposição

Como a eGs abrange três módulos: o pré-processamento, o processamento e o pós-processamento, decidiu-se utilizar o *princípio da decomposição*, que consiste em dividir um problema complexo em subproblemas de solução mais fácil, ficando o problema principal dividido em três partes, que são:

1. Módulo de Pré-processamento: Consiste na edição e armazenamento dos dados do *modelo geoelétrico* e da observação de campo pelo usuário. Nessa fase, o usuário insere ou edita o *modelo geoelétrico* em uma janela e os *dados observados* em outra (Figura 2.3);

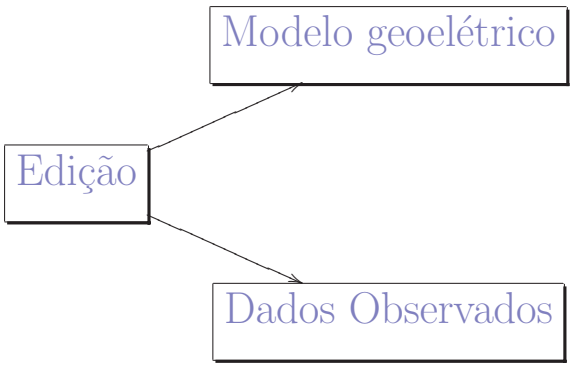


Figura 2.3: Diagrama de pré-processamento dos dados

2. Módulo de Processamento: Consiste na leitura dos arquivos salvos pelo o usuário no pré-processamento e na computação da modelagem numérica. Nessa fase, o computador lê os arquivos editados através de uma janela e cria arquivos de dados diretos ou inversos (Figura 2.4);

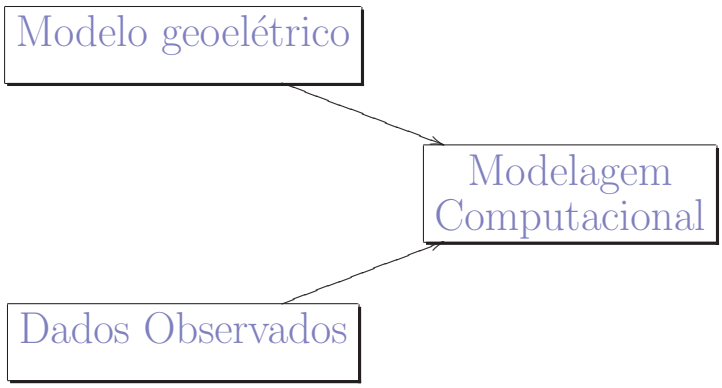


Figura 2.4: Diagrama de processamento dos dados

3. Módulo de Pós-processamento: consiste na leitura dos arquivos criados na modelagem numérica para visualização. Nessa fase, o usuário visualiza gráficos dos arquivos criados (Figura 2.5);

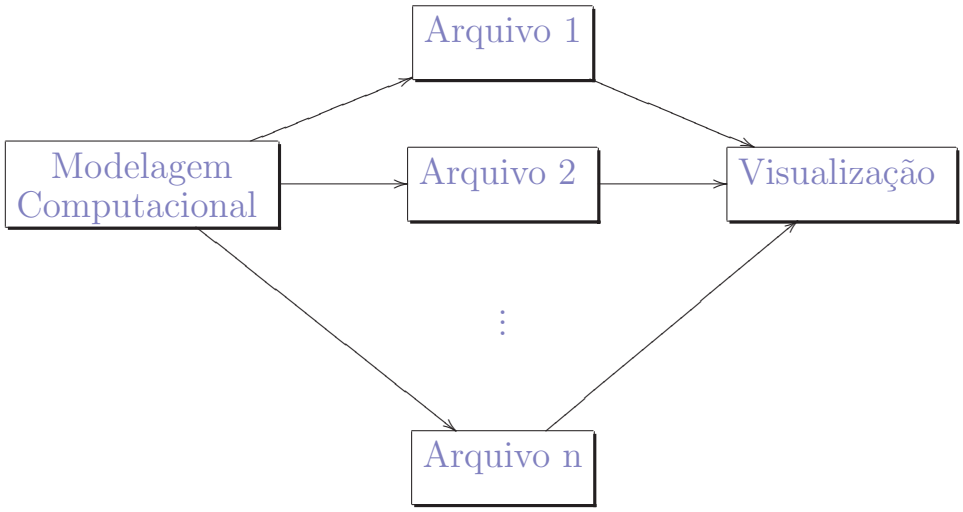


Figura 2.5: Diagrama de pós-processamento dos dados

A primeira parte (pré-processamento) ficou sob a minha responsabilidade. A segunda parte (processamento) são as dlls criadas em FORTRAN 95 pelo Prof. Luiz Rijo. A ultima parte (pós-processamento) foi destinada a Rodrigo E. da Conceição Silva (Silva, R. E. C., 2005).

2.2.3 - Princípio do Refinamento

O princípio do refinamento se baseia em compreender o problema como um todo e avaliar antes as partes mais importantes, para depois dar atenção aos detalhes, pois a execução no sentido contrário pode levar à perda do foco principal do problema.

O problema mais importante é fazer a ligação do usuário diretamente com as dlls, sem precisar adquirir conhecimentos de alguma linguagem de programação. Logo, inicialmente o usuário deve enviar informações para a dll, e para isso é necessário construir meios para que o manipulador do aplicativo tenha acesso aos programas em FORTRAN.

Portanto, para iniciar a construção da eGs foi pensado primeiramente no que o programa deviria ter para ser útil, ou seja, nas opções mais básicas de um aplicativo que satisfizessem as necessidades dos métodos, que seriam implantados posteriormente. Assim, chegou-se às seguintes conclusões:

- É necessário fazer um menu que siga o padrão utilizado pela maioria dos aplicativos do *Windows*, ou seja, com as opções *Arquivo*, *Editar*, *Exibir*, *Formatar*, *Ferramentas* e *Ajuda*, acrescentando-se, além destas, as opções de executar a modelagem e de construção de gráficos para a interpretação;

- É necessário fazer uma janela que o usuário possa fazer as escolhas das dimensões do *modelo geoelétrico* (1D, 2D ou 3D), dos métodos EM e do tipo de problema (direto ou inverso);

- É necessário fazer uma janela pela qual o usuário possa editar o nome dos arquivos que serão salvos e que tenha um botão para executar a modelagem numérica;

- É necessário fazer uma janela para editar os dados que após salvos irão para o arquivo do *modelo geoelétrico* e outra para os *dados observados*.

Inicialmente objetivou-se fazer um aplicativo que englobasse essas opções básicas, no qual seriam acrescentadas mais opções e detalhes, a fim de facilitar a manipulação do *software* e a personalização do aplicativo pelo usuário.

O próximo passo é dar atenção aos detalhes, que são opções fundamentais para o pré-processamento, como salvar e abrir um arquivo, editar o nome do arquivo e executar a modelagem. Assim:

- Para salvar um arquivo existem duas maneiras: a primeira é criar um botão e nele fazer um procedimento para salvar os dados editados em um diretório específico, ou seja, definir um caminho que não poderá ser mudado. Sendo assim, os dados serão salvos um sobre o outro o que não dá a flexibilidade do usuário escolher por conveniência o local para salvar o arquivo. A segunda, a qual foi a implementada, é mostrar a janela de salvar que a maioria dos aplicativos usa para guardar um arquivo, sendo que deste modo os dados serão colocados de acordo com a escolha do usuário.

- Para abrir um arquivo existem dois modos, como na opção salvar, então o raciocínio utilizado foi o mesmo: no primeiro basta criar um botão para abrir um arquivo salvo em um local definido, facilitando desta forma o acesso do usuário aos dados, porém há somente a opção de abrir o último arquivo salvo. Já a segunda maneira é usar a janela modal de abrir arquivo, sendo que assim o usuário pode escolher qualquer arquivo especifico, para a eGs, existente no computador. Neste caso ambos os modos foram necessários implementar.

- Para se ter acesso às dlls, existe somente uma maneira: criar um botão que ative a modelagem numérica. Todavia, as dlls são totalmente transparentes ao usuário.

2.2.4 - Junção das Soluções

Como as aplicações dos métodos elétricos e eletromagnéticos são muitas, decidiu-se fazer, primeiramente, a interface gráfica de cada método para salvar e receber os arquivos dos *modelos geoelétricos* e *dados observados*, para depois fazer testes computacionais para saber se não há nenhum erro lógico e depois testes com as dlls para saber se elas estão bem acopladas ao *software*, pois o paradigma principal do aplicativo é utilizar os programas desenvolvidos em FORTRAN.

3 - A EGS 3.0 E SUA APLICABILIDADE

O objetivo deste capítulo é descrever a funcionalidade e as facilidades do programa eGs para modelagem numérica dos métodos elétricos e eletromagnéticos já implementados na suite. Para isso, primeiramente, será mostrado como usar o executável e posteriormente será apresentado o pré-processamento, processamento e pós-processamento de algumas aplicações geofísicas em 1D e 2D. As aplicações 3D são relatadas em Silva, R. E. C. (2005).

3.1 - CONHECENDO A eGs

A eGs é um programa desenvolvido, exclusivamente, para modelagem numérica de métodos elétricos e eletromagnéticos. A nova versão (eGs 3.0) teve uma reestruturação na arquitetura, pois o programa desenvolvido anteriormente (eGs 2000), apesar de ter resolvido o problema do geofísico não necessitar de uma linguagem de programação para poder usar os pacotes da EGSLIB (Electrical Geophysics Software LIBrary), primeira versão da eGs, não solucionou a facilidade de escolha da dimensão (1D, 2D ou 3D), de escolha dos métodos, da criação de novos *modelos geoelétricos* e da criação de novos *dados observados*. Além disso, na versão anterior, para abrir arquivos salvos também não era uma tarefa rápida.

As manipulações na eGs 2000 se tornaram difíceis, pois toda funcionalidade do programa estava concentrada no *menu*. É sabido que atualmente um aplicativo deve ter lugares diferentes para o usuário fazer a mesma atividade no programa, por exemplo, a opção *Abrir*, em muitos programas, não está apenas no menu, mas também em outras janelas do mesmo aplicativo.

As atualizações mais importantes na eGs, são aquelas em que afetam diretamente a interatividade do usuário com o computador, ou seja, para deixar o acesso ao sistema operacional, através do programa, mais amistoso. Mudando, portanto:

- O visual;
- As manipulações das janelas;
- Os posicionamentos dos botões;
- Informações de uso da eGs;
- Arquivo de instalação;

- Distribuição da eGs;

O visual de um programa é um fator importante para transmitir segurança e conforto. A estética de um aplicativo está diretamente ligada ao gosto particular, mas um programa que se assemelha aos padrões atuais da Microsoft, tem a possibilidade maior do usuário se adaptar ao ambiente e, conseqüentemente, sentir-se mais seguro e confortável à utilização do mesmo. Portanto, a eGs seguiu os padrões da última versão do *Windows*, o *Windows XP*, e as cores, para distinguir procedimentos, foram usadas com tons de poucas variações.

As janelas foram feitas de modo que uma fosse independente da outra para ser movimentada, tendo como limites apenas a tela do computador. Portanto, foram criadas *janelas sobrepostas*.

O posicionamento dos botões foi colocado em lugares estratégicos para facilitar a sua localização e para que alguns procedimentos da eGs serem executados apenas quando outras ações tenham sido praticadas anteriormente pelo usuário. Por exemplo, o botão *Executar* só pode ser clicado quando a janela "Modelo" e "Dados observados" estiverem abertas, pois para fazer a modelagem numérica, é necessário os arquivos do *modelo geoelétrico* e dos *dados observados*.

Informações de uso da eGs são transmitidas, também, durante as atividades do usuário com o executável, ou seja, caso a pessoa faça alguma etapa errada durante o processo de modelagem, o aplicativo informa a manipulação correta que deve ser feita. Dessa forma, o interessado vai aprendendo a usar a eGs sem ter que, em alguns casos, parar e ler o Manual.

Para facilitar a distribuição e obtenção da eGs 3.0, preparou-se um arquivo de instalação do aplicativo, que possui todos arquivos necessários ao bom funcionamento do programa em apenas um pacote. Desse modo o usuário não precisa ter nenhum programa de compactação de arquivos instalado no computador, evitando, com isso, cometer o erro de salvar a eGs em diretório não aceito.

3.1.1 - Distribuição

A eGs 3.0 e a primeira versão do Manual, que é o Apendice A deste trabalho, serão distribuídos através do *site* do Prof. Luiz Rijo cujo endereço é *www.rijo.pro.br*.

O Manual, assim como a eGs, serão periodicamente atualizados com sugestões dos usuários e com implementação de novos métodos àqueles não ativados na eGs 3.0. É importante informar que a eGs é um software livre, ou seja sem fins lucrativos e , portanto, a colaboração

dos usuários é de suma importância para melhorar o desempenho do aplicativo.

3.1.2 - Instalação

A eGs 3.0 deve ser instalada sempre no seguinte diretório:

C:\ eGs

Este caminho não deve ser alterado, pois o programa é dependente de arquivos complementares. Os arquivos complementares, são dados em que a eGs usa para saber onde os arquivos do *modelo geoelétrico* e dos *dados observados* foram salvos pelo usuário ou pelo próprio programa. Além disso, as bibliotecas de ligação dinâmica que executam a modelagem numérica, funcionam apenas com os arquivos auxiliares em locais específicos.

3.1.3 - Inicialização do Programa

Após a instalação da eGs, o programa está pronto para ser usado. Para ter acesso ao aplicativo eGs, basta ir em C:\eGs e será encontrado o seu ícone (Figura 3.1), o qual é possível inicializar o programa.

Figura 3.1: Ícone da eGs 3.0

Após ter inicializado a eGs, aparecerá uma imagem dinâmica de quatro segundos, que é a tela de apresentação da eGs (Figura 3.2).

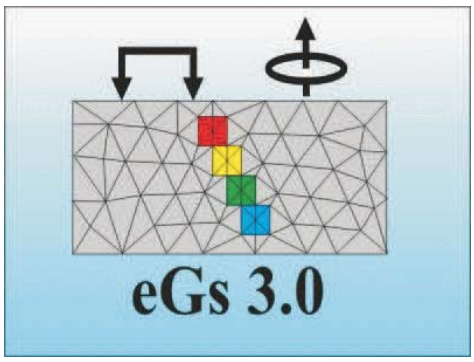


Figura 3.2: Tela de apresentação da eGs 3.0

Observa-se em seguida três janelas que compõem a tela principal do programa, como mostra a Figura 3.3. Assim, ao iniciar o funcionamento da eGs pode ser visto as janelas:

- **eGs 3.0**: parte superior da tela;
- **Métodos**: lado esquerdo superior;
- **Seleção**: lado esquerdo inferior.

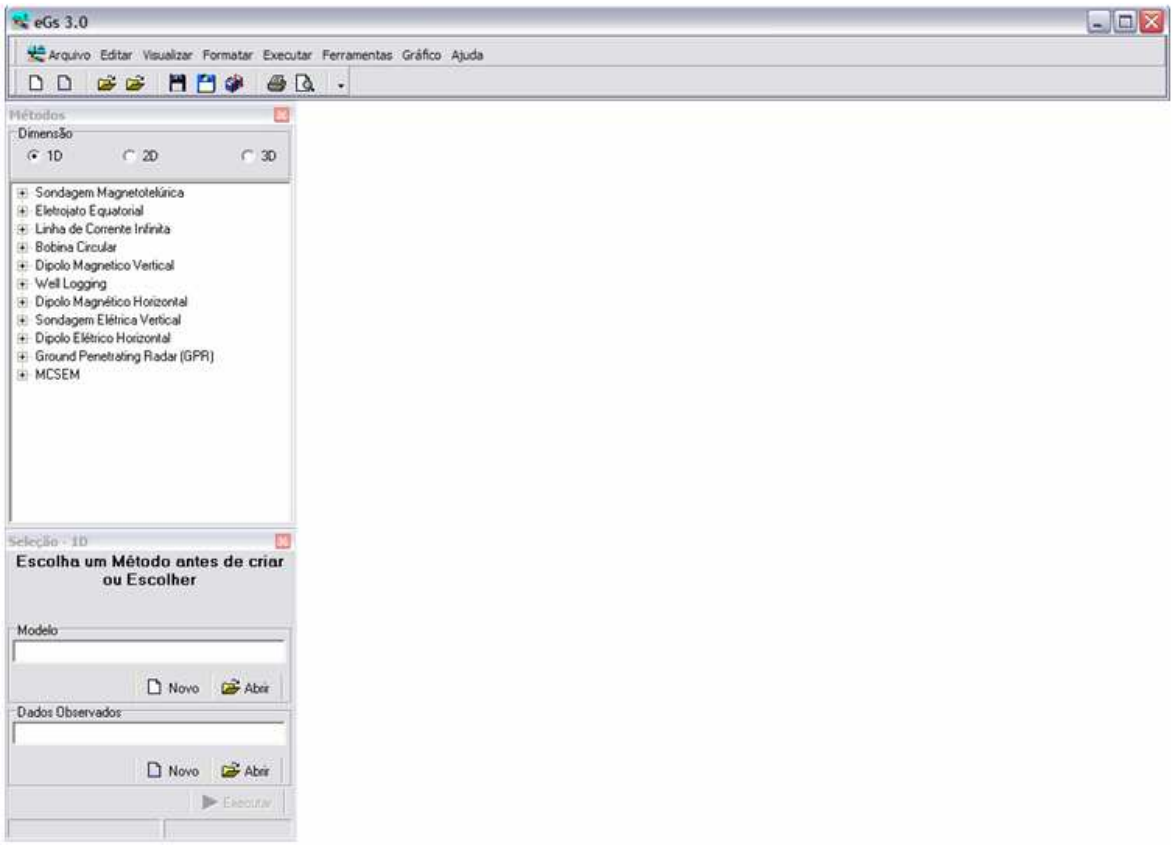


Figura 3.3: Aplicativo eGs 3.0

O usuário pode posicionar as janelas “eGs 3.0”, “Métodos” e “Seleção” em qualquer lugar na tela do computador, inclusive uma pode sobrepor a outra.

3.1.4 - Janela “eGs 3.0”

A janela “eGs 3.0” (Figura 3.4) é formada por um menu e uma barra de ferramentas, onde o usuário tem acesso a opções para a manipulação dos arquivos e pode fechar ou abrir

todas as janelas existentes no programa. Os aplicativos do Windows possuem uma série de diretrizes padrão do ambiente operacional que descreve como organizar os comandos em um menu. Basta observar os menus de alguns aplicativos mais difundidos, não sendo, portanto, uma invenção dos programadores.

Um menu suspenso deve ter a opção *Arquivo*, seguido por *Editar*, *Exibir(ou Visualizar)* e depois por alguns determinados comandos específicos do aplicativo. A parte final da seqüencia inclui *Opções*, *Ferramentas* e sempre termina com *Ajuda*. Cada um destes itens do menu suspenso tem uma distribuição padrão, embora os itens reais dependam do aplicativo. Por exemplo, o item *Arquivo* costuma ter comandos tais como *Novo*, *Abrir*, *Salvar*, *Salvar Como*, *Imprimir* e *Sair*.

Os itens e subitens do menu têm um outro recurso padrão: as teclas de atalho (*short cut*). Quando é visto a primeira letra de um item ou subitem sublinhada, ou uma descrição resumida de uma tecla ou combinação de teclas, junto com um item do menu, significa que é possível pressionar essas teclas para obter o comando relacionado, por exemplo, para acessar o item *Arquivo* da eGs 3.0, basta pressionar, simultaneamente Alt e a tecla "A"(atalho: Alt+A) . Embora emitir comando pelo mouse seja muito fácil, isto costuma ser lento, principalmente para aplicativos que requerem muito o teclado, pois uma das mãos deve ser deslocada do teclado ao mouse.

Figura 3.4: Janela eGs 3.0

Outra facilidade que a janela "eGs 3.0" possui é a escolha do usuário arrastar o menu e a barra de ferramentas dentro da janela ou fora dela, como pode ser visto na Figura 3.5.

Figura 3.5: Menu da eGs 3.0 com a barra de ferramentas fora da janela principal

A janela principal do aplicativo é a "eGs 3.0", ou seja, todas atividades no executável podem ser feitas nela, mas não exclusivamente. Portanto, se a janela "eGs 3.0" for fechada

o usuário está optando para sair do aplicativo.

A função dos itens do menu é armazenar subitens que possuem procedimentos específicos. Assim:

- **Arquivo**: Este item possui os seguintes subitens:
  - **Novo**
    - **Modelo:** Este Botão tem a função de abrir uma janela "Modelo" vazia para edição de dados para o *modelo geoelétrico*;
    - **Dados Observados:** Este Botão tem a função de abrir uma janela "Dados Observados" vazia para edição de *dados observados* de campo.
  - **Abrir**
    - **Modelo:** Este Botão tem a função de abrir uma janela "Modelo" com um arquivo do *modelo geoelétrico* escolhido pelo usuário;
    - **Dados Observados:** Este Botão tem a função de abrir uma janela "Dados Observados" com um arquivo de *dados observados* de campo escolhido pelo usuário.
  - **Fechar:** Este Botão tem a função de fechar a última janela "Modelo" ou "Dados Observados" que está sendo usada;
  - **Fechar Tudo:** Este Botão tem a função de fechar todas janelas "Modelo" e "Dados Observados" que estão sendo usadas;
  - **Salvar Modelo:** Este Botão tem a função de armazenar em um arquivo, com extensão *.MGE, os dados editados na janela "Modelo";
  - **Salvar Dados Observados:** Este Botão tem a função de armazenar em um arquivo, com extensão *.OBS, os dados editados na janela "Dados Observados";
  - **Salvar Tudo:** Este Botão tem a função de armazenar os dados editados na janela "Modelo" e "Dados Observados".
  - **Sair:** Este Botão tem a função de fechar o aplicativo eGs 3.0. Caso os dados editados na janela "Modelo" ou "Dados Observados" não tenham sidos salvos, o programa mostra uma janela solicitando a opção de armazenar as atualizações.
- **Editar**:

  - **Recortar:** Copia o texto selecionado para ser usado em outro lugar e deixa vazio o local onde o texto estava;
  - **Copiar:** Apenas copia o texto selecionado para ser usado em outro lugar, não deixando em branco a seleção;
  - **Colar:** Insere, na posição do cursor, um texto recortado ou copiado;
  - **Excluir:** Apaga o texto selecionado;
  - **Desfazer:** Retorna à última ação;
  - **Refazer:** Avança à última ação desfeita;

- **Visualizar**:
  - **Árvore de Métodos:** Fecha ou abre a janela "Métodos";
  - **Seleção:** Fecha ou abre a janela "Métodos".

- **Formatar**:
  - **Fonte**: Muda a fonte do texto.

- **Executar**:
  - **Executar com Gráfico**: Tem a função de efetuar a modelagem numérica e mostrar a figura do resultado;
  - **Executar sem Gráfico**: Tem a função de efetuar a modelagem numérica, não mostrando a figura do resultado;

- **Gráfico**: As informações sobre todos os comando deste item pode ser obtidos no Apêndice A deste documento.
  - **Curvas**;
    - **Abrir Arquivo**;
    - **Limpar**;
    - **Quantidade**;
      - **1 Gráfico**
      - **2 Gráficos**
        - **Juntos**;
        - **Separados**.

            - **4 Gráficos**
                - **Juntos**;
                - **Separados**.
        - **Escala**;
        - **Escolher Colunas**;
        - **Alterar Limites**;
            - **Linear-Linear**;
            - **Log-Linear**;
            - **Log-Log**;
            - **Limites Automáticos**.
        - **Identificação**;
        - **Símbolos**;
        - **Gerar Curvas**;
            - **Sem Malha**;
            - **Malha Parcial**;
            - **Malha Completa**.
    - **Mapas**;
        - **Abrir Arquivo**;
        - **Escolher Arquivos MCSEM**.
    - **Pseudo-seções do IP**.
        - **Escolher Arquivos IP**.

- **Ajuda**:
    - **Ajuda:** Tem a função de mostrar uma janela de informações de uso da eGs 3.0;
    - **Sobre:** Tem a função de mostrar uma janela de informações autorais do aplicativo.

A janela “eGs 3.0” possui, também, uma barra de ferramentas que fica localizada abaixo do menu, cuja finalidade é mostrar os comandos mais usados no menu, de modo a agilizar a escolha pelo usuário.

Os dados editados na janela “Modelo” ou “Dados Observados” podem ser salvos usando os botões “Salvar Modelo” ou “Salvar Dados Observados”, os quais mostram uma janela modal (Figura 3.6), Cantù, M. (2002), para o usuário salvar o arquivo no local desejado.

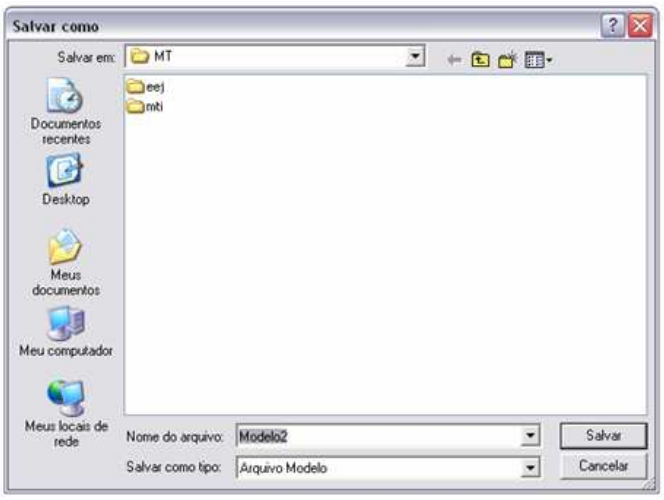


Figura 3.6: Janela modal para salvar arquivos

Observa-se, portanto, que o menu e a barra de ferramentas possuem a estrutura comum à maioria dos aplicativos desenvolvidos para *Windows*, facilitando portanto a adaptação do usuário com a eGs.

3.1.5 - Janela "Métodos"

Os métodos elétricos e eletromagnéticos formam uma grande quantidade de técnicas de exploração geofísica. Para cada método existe o problema direto e o problema inverso. O problema direto tem o objetivo de simular *dados observados*, a partir dos parâmetros de um *modelo geoelétrico* dado, já o problema inverso tem o objetivo de determinar uma estimativa dos parâmetros do *modelo geoelétrico* a partir das observações, que podem ser dados de campo ou dados simulados no problema direto. As aplicações geofísicas são divididas pelas dimensões da geometria do *modelo geoelétrico*: 1D, 2D e 3D.

Por isso, a janela "Métodos" (Figura 3.7) tem a finalidade de disponibilizar ao usuário a escolha, a partir da dimensão, o método que se deseja usar no problema direto ou inverso. No momento, nem todos os métodos estão ativados, pois a eGs é ampla e de longa duração de desenvolvimento, portanto muitos outros métodos serão ativados nas futuras versões. A lista hierárquica gráfica de métodos é fundamental para facilitar a escolha do método EM e o usuário que estiver utilizando o aplicativo não ultrapasse nenhuma das etapas do processo de escolha, pois primeiramente deve ser determinada a dimensão, depois a técnica e por último o tipo de problema. A hierarquia dos métodos na janela "Métodos" foi feita de acordo com ordem posta em Rijo (1992).

Figura 3.7: Árvore de Métodos da eGs 3.0

O símbolo $\boxed{+}$ antes do nome do método indica que existe subitem para selecionar e o sinal de $\boxed{-}$, quando a árvore está expandida. Portanto, com esta janela é possível iniciar a edição e abrir arquivos de *modelos geoelétricos* e *dados observados* do método desejado, porém não mostrará os formulários para edição ou reedição, pois a árvore ficaria com muitas opções e fugiria do objetivo deste trabalho, que é facilitar a funcionalidade do usuário com o programa.

3.1.6 - Janela "Seleção"

A janela "Seleção" é a união de todos os comandos que o usuário pode fazer após a escolha do problema direto ou inverso, facilitando o acesso à escolha do tipo de arquivo, os quais se deseja editar ou abrir, e simplificando a execução da modelagem numérica. Todos os botões existentes em "Seleção", também estão no menu, mas por questão de comodidade para o usuário, adotou-se a estrutura mostrada na Figura 3.8.

Como é possível observar, a janela "Seleção" mostra em seu título a dimensão que está sendo usada e na parte superior o método escolhido, como exemplo, a Figura 3.8 mostra que o método escolhido na janela "Métodos" foi o magnetotelúrico isotrópico. As caixas de edição servem para nomear os arquivos de *modelos geoelétricos* e *dados observados*. Os botões *Novo*, *Abrir* e *Executar* produzem os mesmos comandos existentes no menu, sendo possível abrir várias janelas para serem editadas ao mesmo tempo.

Figura 3.8: Janela Seleção da eGs 3.0

Se os botões *Novo* ou *Abrir* forem clicados antes de fazer a escolha do método EM na janela "Métodos", o usuário será informado de que é necessário escolher um método, como mostra a Figura 3.9.(a). Caso o usuário escolha o método e não indique o tipo de problema, ocorre o mostrado na Figura 3.9.(b).

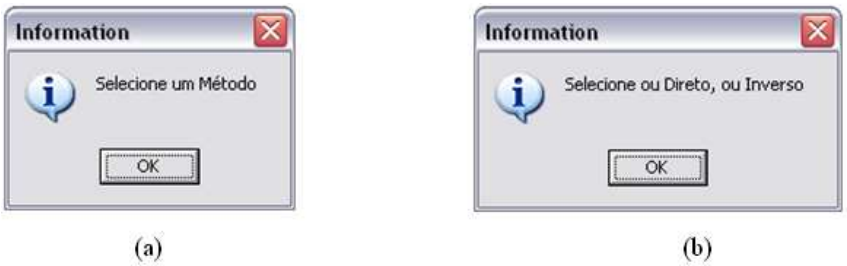


Figura 3.9: (a)Janela que informa para selecionar um método (b)Janela que informa para selecionar o tipo de problema (direto ou inverso).

O botão *Executar* é ativado apenas quando está aberta, pelo menos, uma janela "Modelo" e uma "Dados Observados". Após esse botão ativado, o usuário pode fazer o processamento para a criação de arquivos do problema direto ou inverso, e os gráficos com os arquivos de saída para a interpretação geofísica.

A vantagem da janela "Seleção" é que nela existem todos os procedimentos para iniciar o pré-processamento e processamento do método escolhido na janela "Métodos". O pós-processamento, também pode ser efetuado na janela "Seleção" através do botão *Executar* se o item *Executar* do menu estiver selecionado *Executar com Gráfico*, pois nele se gera o gráfico para interpretação, mas usando opções padronizadas pela eGs. Caso o usuário escolha *Executar sem Gráfico*, apenas serão criados os arquivos de saída e não será mostrado nenhuma figura à interpretação.

3.1.7 - Janelas "Modelo"

Na aplicação dos métodos geofísicos elétricos e eletromagnéticos, procuram-se *modelos geoelétricos* teóricos que reproduzam os *dados observados* no campo. Isto pode ser feito de duas maneiras: resolvendo o problema direto, cujo objetivo é cálcular os *dados observados* tendo conhecimento dos parâmetros do *modelo geoelétrico*; e resolvendo o problema inverso, o qual é obter uma estimativa dos parâmetros do *modelo geoelétrico* a partir das observações.

A janela "Modelo", na eGs, tem a função de editar os dados dos *modelos geoelétricos*. Para representar o *modelo geoelétrico* teórico em camadas estratificadas, os métodos aplicados para uma dimensão usa a janela padrão (Figura 3.10.(a)) com a opção *permeabilidade* (Figura 3.10.(b)) para editar a permeabilidade magnética relativa do meio, que na maioria dos métodos eletromagnéticos será igual a um, salvo quando existem rochas ricas em minerais magnéticos (magnetita, ilumenita, etc). Os métodos MCSEM (*Marine Controlled Source ElectroMagnetic*) e as sondagens elétricas verticais Schlumberger, Wenner e Dipolo-Dipolo não usam a permeabilidade relativa, portanto opção *permeabilidade* é desativada.

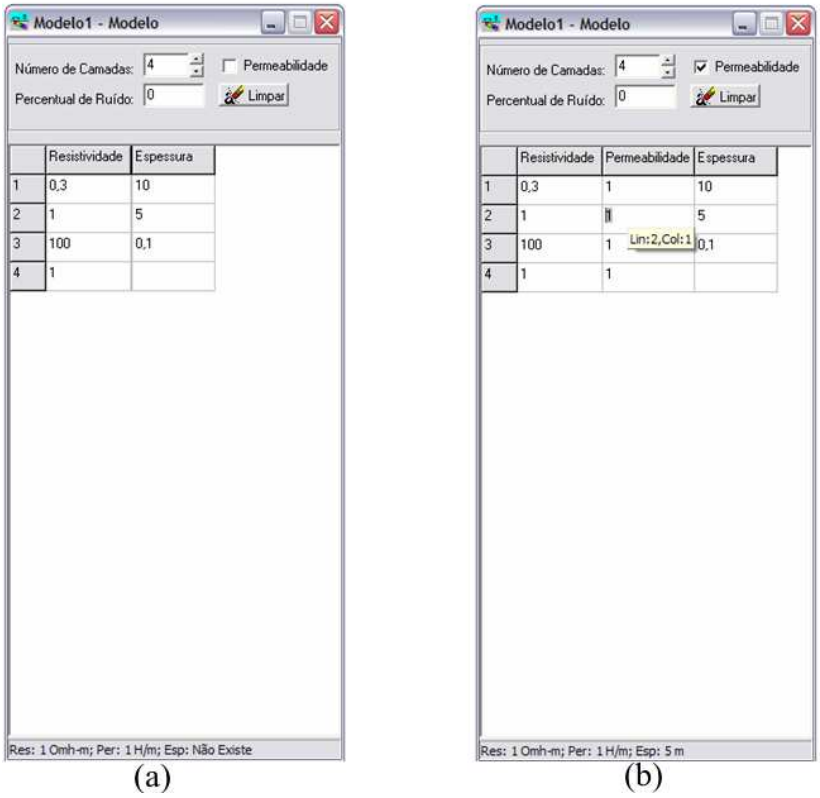


Figura 3.10: (a) Janela padrão do *modelo geoelétrico* 1D. (b) Janela "Modelo" com a opção *permeabilidade* selecionada para alterar o valor.

O número de camadas nos *modelos geoelétricos* não precisam ser grande, portanto o valor máximo para *Número de Camadas* é cem. Caso o valor editado ultrapasse cem camadas, é mostrada uma janela modal informando o erro do usuário, como mostra a Figura 3.11.

Figura 3.11: Janela informativa para o número de camadas.

As linhas da tabela são acrescentadas através do valor no *Número de Camadas*, onde a primeira coluna, que não pode ser editada, é reservada para numerar as camadas, a segunda para a resistividade, a terceira para permeabilidade e a última para a espessura.

Naturalmente, o substrato, que é a última camada, não tem espessura, portanto a célula referente a última espessura não está habilitada à edição. Se o usuário tentar editar algum valor, é mostrada a janela modal 3.12 informando que não existe espessura.

Figura 3.12: Janela informativa ao editar a última espessura.

3.1.8 - Janelas "Dados Observados"

Os *dados observados* nos métodos elétricos e eletromagnéticos correspondem às medidas obtidas no campo por prospectores ou, em alguns casos, são dados sintéticos gerados no computador, com a finalidade de caracterizar, de forma quantitativa, o meio geológico de acordo com as suas propriedades elétricas.

A janela "Dados Observados", tem a função de ler, editar e salvar arquivos que representam os *dados observados* de campo. Os métodos não possuem uma estrutura uniformizada à janela "Dados Observados", mas há duas formas de edição: por tabelas e por pseudo-seções. O tipo tabela é semelhante ao descrito anteriormente pela janela "Modelo". O tipo

de pseudo-seção é representação das mediadas em posições aparentes da subsuperfície.

Todos os métodos em 1D usam a janela "Dados Observados" na forma de tabela. Alguns métodos 2D e 3D também usam esse formato, por exemplo o método VLF (Very Low Frequency) que é um método em 2D. A janela mais simples para os *dados observados* na forma de tabela é da SEV (Sondagem Elétrica Vertical), como é possível ver na Figura 3.13.(a) e a mais complicada, ou seja, exige maior quantidade de valores serem inseridos é a simulação do eletrojato equatorial em sondagens magnetotelúricas (Figura 3.13.(b)).

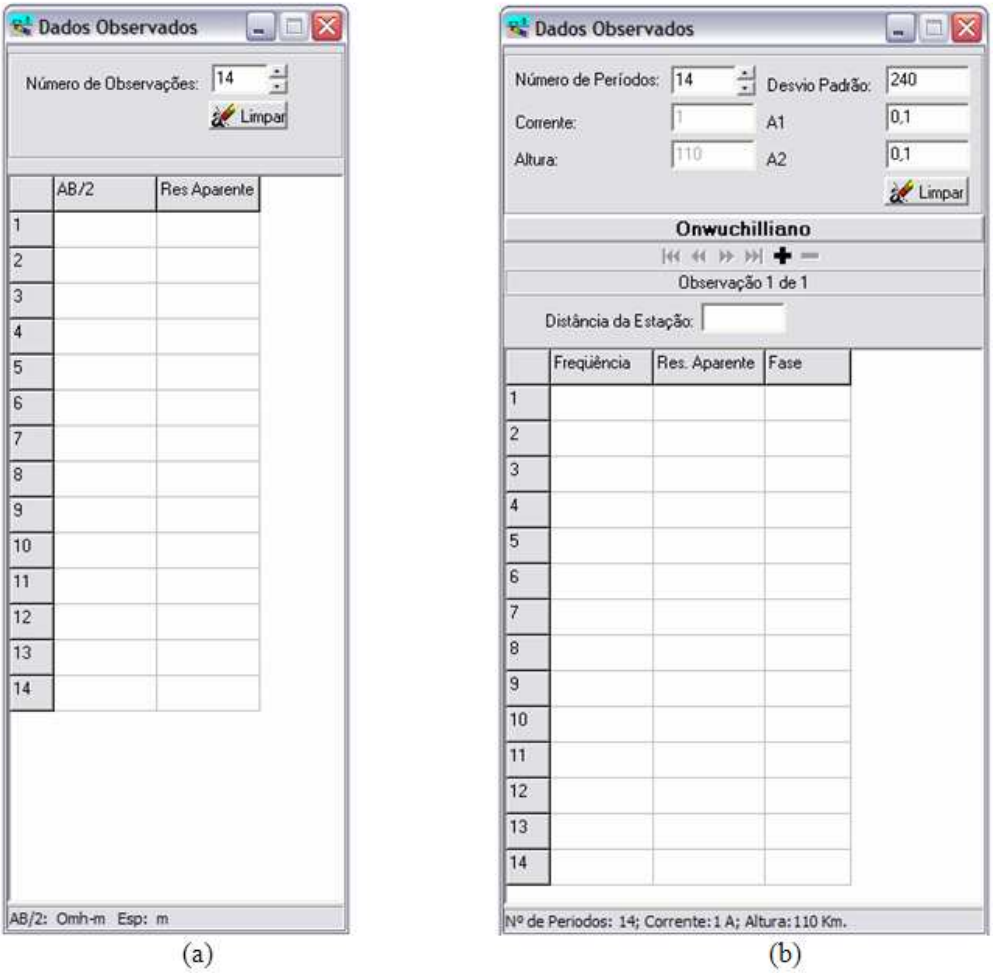


Figura 3.13: (a)Janela para os *dados observados* da SEV (b)Janela para os *dados observados* da sondagem magnetotelúrica para simular o eletrojato equatorial.

No caso magnetotelúrico com eletrojato equaltorial, é importante citar que na janela "Dados Observados" a corrente e a altura não são alteradas, mas são mostradas, pois um dos motivos da criação da eGs é instruir alunos de graduação. Os valores dos parâmetros *desvio*

padrão, *A1* e *A2* alteram a estrutura do eletrojato equatorial, ou seja:

- **Linha Infinita:** Quando for editado 0 para *desvio padrão*, 0 para *A1* e 0 para *A2* o eletrojato equatorial se comporta com uma linha infinita de corrente.

$$\begin{cases} \textit{desvio padrão}=0; \\ \textit{A1}=0; \\ \textit{A2}=0. \end{cases}$$

- **Gaussiano:** Quando for editado valor diferente de 0 para *desvio padrão*, 0 para *A1* e 0 para *A2* a densidade de corrente se comporta como uma distribuição gaussiana. A eGs usa como padrão os seguintes valores:

$$\begin{cases} \textit{desvio padrão}=240; \\ \textit{A1}=0; \\ \textit{A2}=0. \end{cases}$$

- **Onwuchilliano:** Quando for editado valores diferentes de 0 para *desvio padrão*, para *A1* e *A2*, estará sendo considerado os contra-eletrojatos equatoriais. A eGs usa como padrão os seguintes valores:

$$\begin{cases} \textit{desvio padrão}=240; \\ \textit{A1}=0{,}1; \\ \textit{A2}=0{,}1. \end{cases}$$

A estrutura de pseudo-seções da janela "Dados Observados", é para, leitura, edição e armazenamento de dados da Polarização Induzida e Resistividade (RES-IP), onde número de níveis é metade do número de eletrodos menos um. Os valores no RES-IP são editados de acordo com as propriedades físicas: resistividade aparente (ρ_a), percentual do efeito da freqüência (PFE - Percentual Frequency Effect) e o fator metálico (MF - Metal Factor). Como mostra a Figura 3.14.

3.2 - EXEMPLOS

Esta seção tem a finalidade de mostrar a facilidade de uso da eGs e seu desempenho em aplicações geofísicas que utilizam os métodos elétricos e eletromagnéticos. Outros exemplos serão mostrados no Manual, que é o Apêndice A desta dissertação.

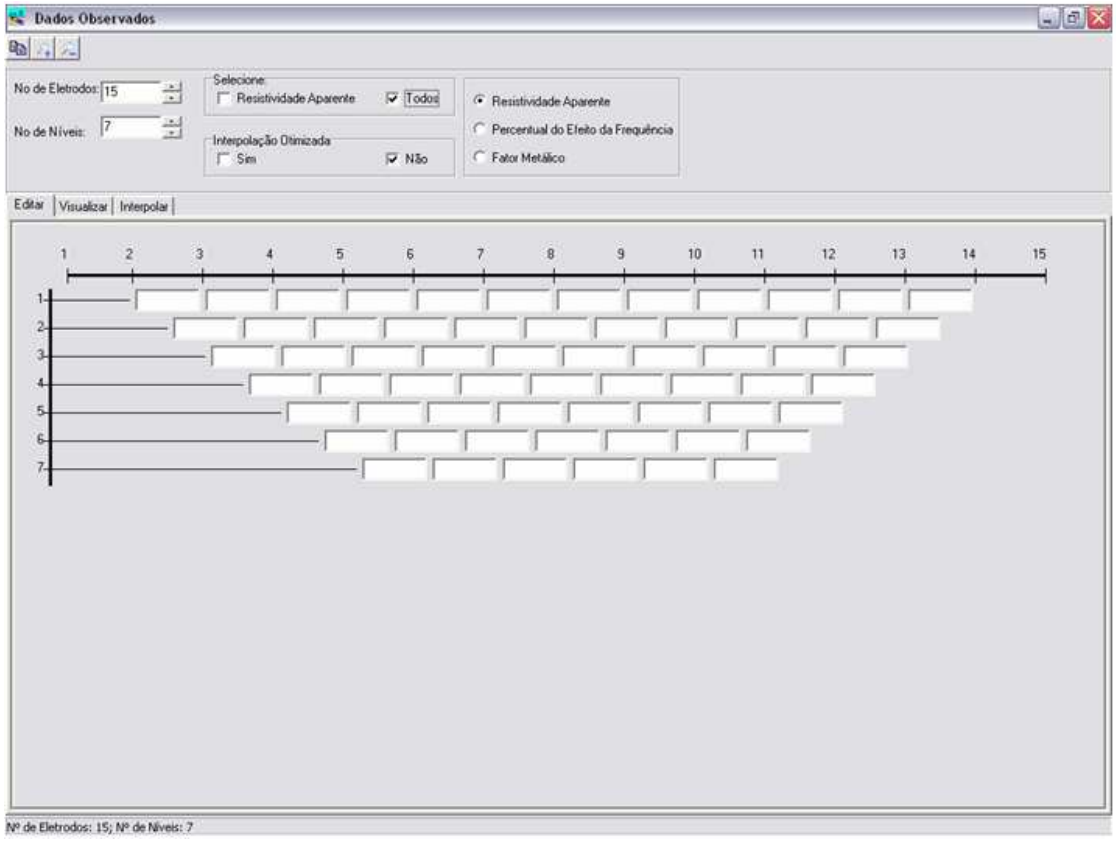


Figura 3.14: Janela para os *dados observados* de IP e Resistividade.

3.2.1 - 1D - Uma dimensão

Para ilustrar a eGs em atividades geofísicas 1D, escolheu-se a sondagem magnetotelúrica isotrópica retirada de Rijo (1992), onde o *modelo geoelétrico* considerado é constituído por cinco camadas distribuídas como mostra a Figura 3.15 e os *dados observados* foram obtidos com 26 freqüencias (0.01, 0.01585, 0.02512, 0.03981, 0.06309, 0.1, 0.1585, 0.2512, 0.3981, 0.6309, 1, 1.585, 2.512, 3.981, 6.309, 10, 15.85, 25.12, 39.81, 63.09, 100, 158.5, 251.2, 398.1, 630.9, 10000)(Figura 3.15).

Os arquivos do *modelo geoelétrico* e *dados observados* foram nomeados, respectivamente, por "MgeMT1d" e "ObsMT1d", os quais são parte integrante do pacote de instalação da eGs, geraram, pela eGs 3.0 através do botão *Executar* da janela "Seleção", as curvas de resistividade aparente e fase, como mostra a Figura 3.16.

Para mostrar que o usuário pode confiar no processamento para a interpretação, construiu-se os gráficos 3.16.(a) e 3.16.(b), sem ruído. E as curvas 3.16.(c) e 3.16.(d) foram geradas,

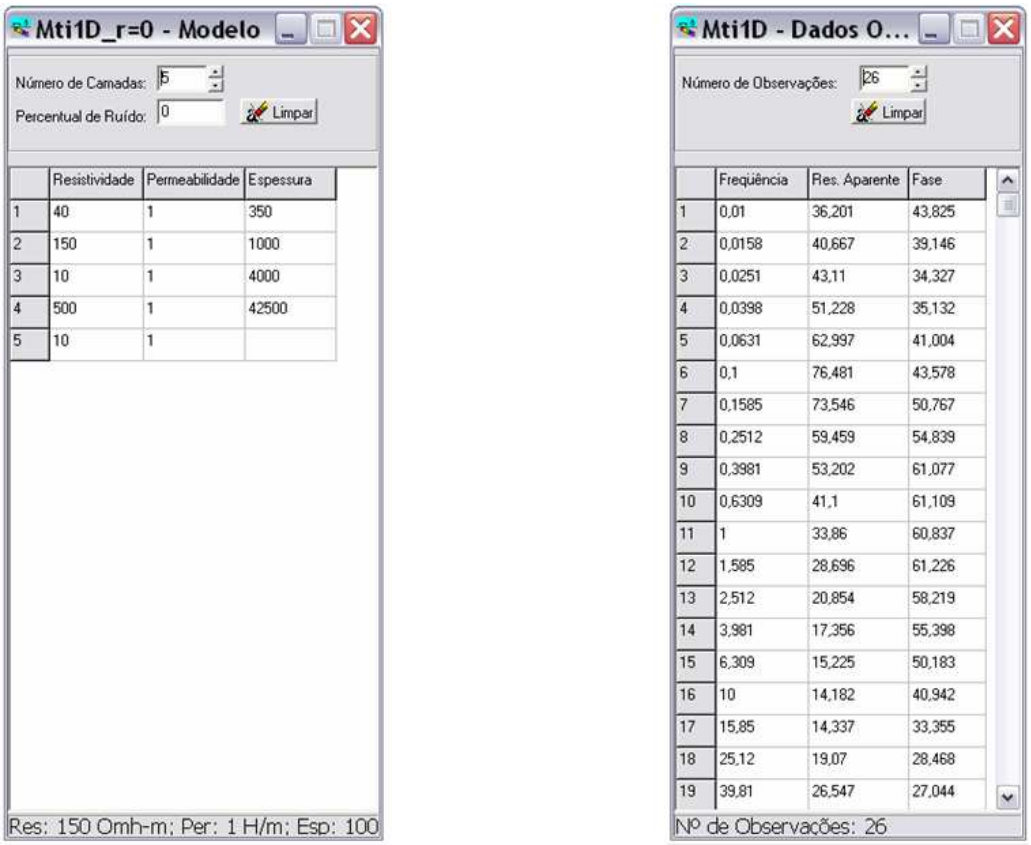


Figura 3.15: (a)*modelo geoelétrico* com 5 camadas assim distribuídas: 40 Ω-m (350 m), 150 Ω-m (1000 m), 10 Ω-m (4000 m), 500 Ω-m (42500 m) e 10 Ω-m; (b)*dados observados* em 26 freqüencias.

admitindo 3% de ruído na resposta do *modelo geoelétrico*.

Os arquivos 'MgeMT1d.MGE" e "ObsMT1d.OBS", respectivamente, *modelo geoelétrico* e *dados observados*, tornaram-se parte integrante do pacote de instalação, para titulo de exemplo, podendo ser encontrados em:

C:\eGs\Exemplos\1D\MT.

3.2.2 - 2D - Duas dimensões

O exemplo de demonstração da utilidade da eGs 3.0, em duas dimensões, é calcular as pseudo-seções de resistividade e polarização induzida obtidos com arranjo bipolo-bipolo para o modelo de um corpo bidimensional mineralizado (Rijo, 1992). Para isso, foi usado o arquivo

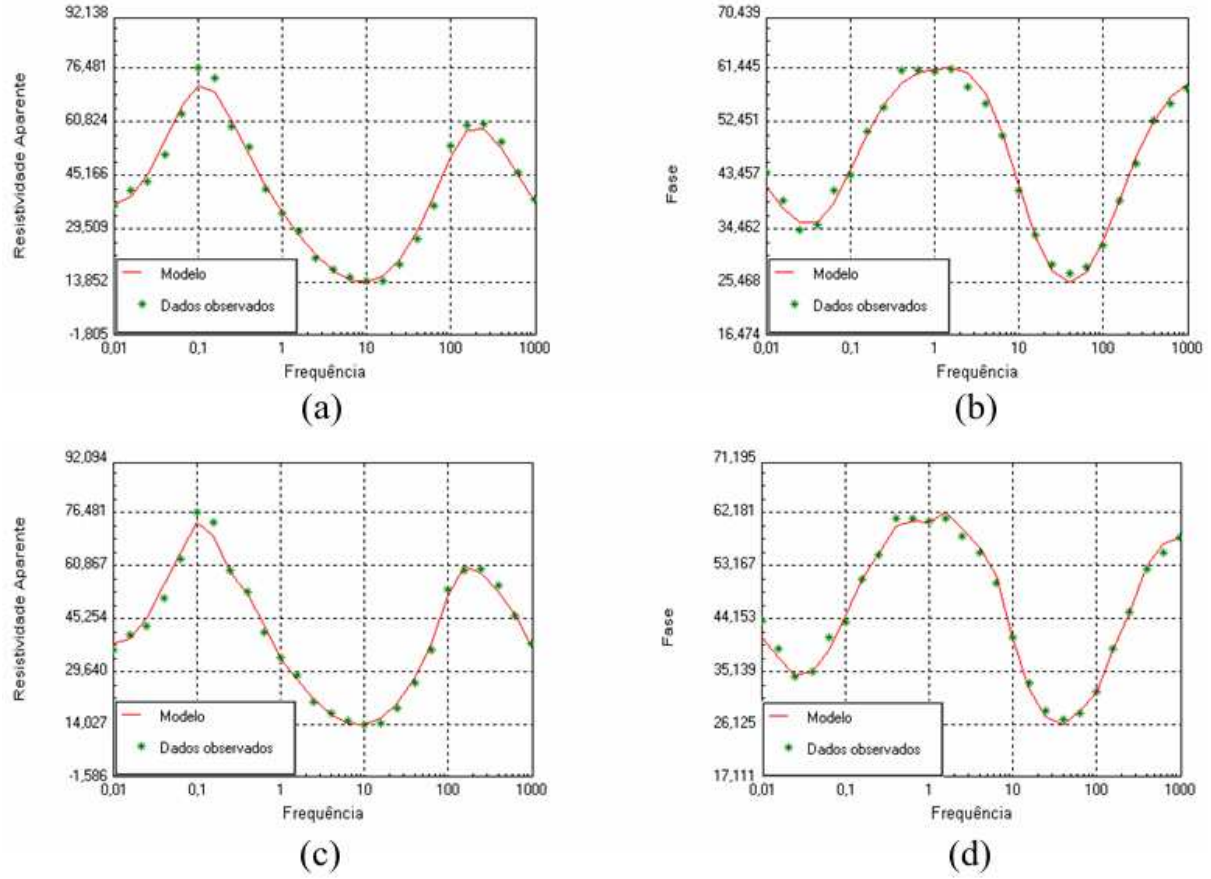


Figura 3.16: (a) Curva da resistividade da resposta do *modelo geoelétrico* "MgeMT1d" com 0% de ruído e dos *dados observados* "ObsMT1d". (b) Curva da fase da impedância da resposta do *modelo geoelétrico* "MgeMT1d" com 0% de ruído e dos *dados observados* "ObsMT1d". (c) Curva da resistividade da resposta do *modelo geoelétrico* "MgeMT1d" com 3% de ruído e dos *dados observados* "ObsMT1d". (d) Curva da fase da impedância da resposta do *modelo geoelétrico* "MgeMT1d" com 3% de ruído e dos *dados observados* "ObsMT1d".

"resIp2D181x18" para *modelo geoelétrico* em malha de elementos finitos, cujas propriedades elétricas podem ser vistas na Figura 3.17.

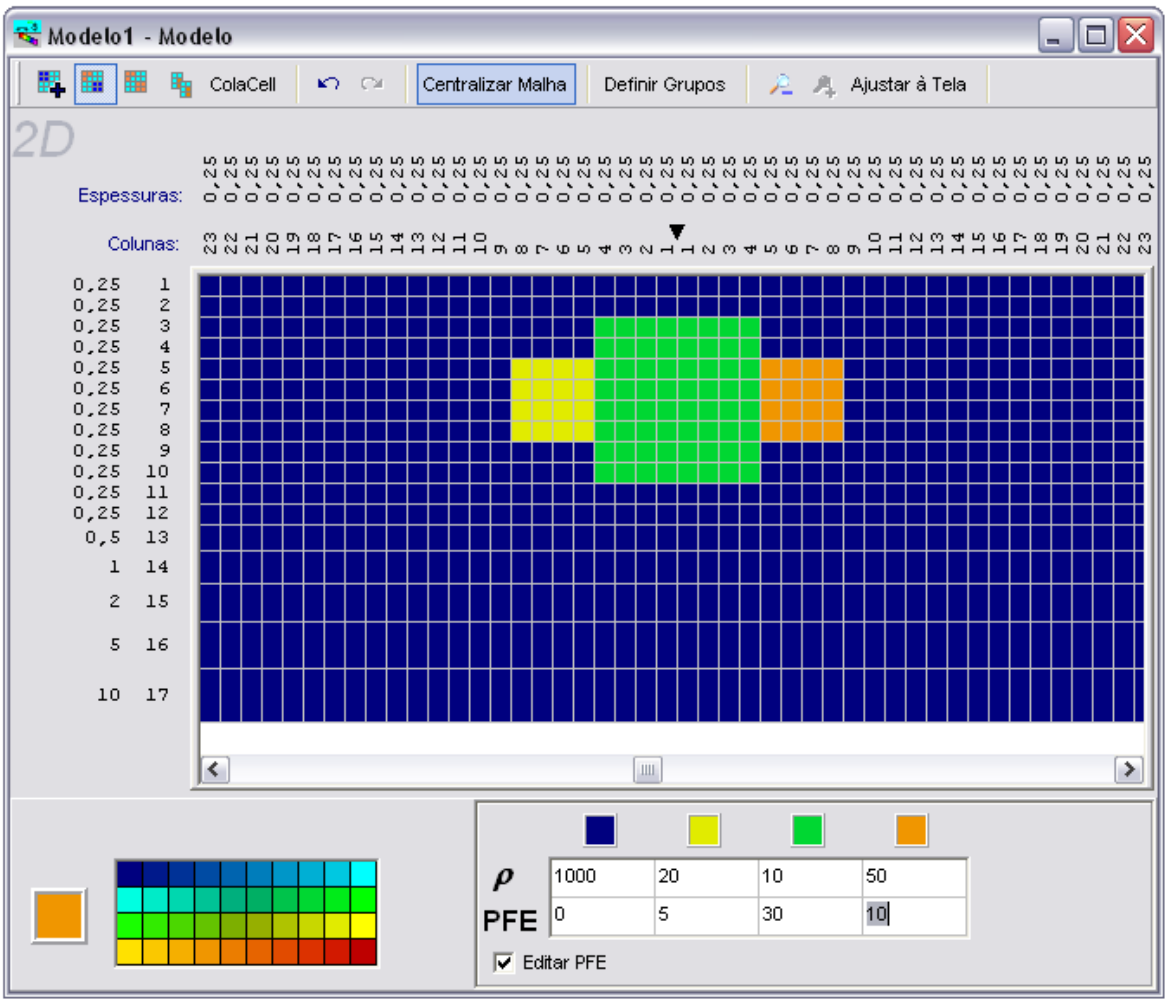


Figura 3.17: Malha do *modelo geoelétrico* "resIp2D181x18".

Os *dados observados* foram editados na Figura 3.14 mostrada na página 30, mas com vinte e cinco eletrodos em até doze níveis de resistividade aparente, PFE e fator metálico, onde os contrastes de valores das propriedades podem ser vistos através da seção *Visualizar* da janela "Dados Observados" para o IP (Figuras 3.18.(a), 3.19.(a) e 3.20.(a)) e também em *Interpolação Otimizada* (Figuras 3.18.(b), 3.19.(b) e 3.20.(b)).

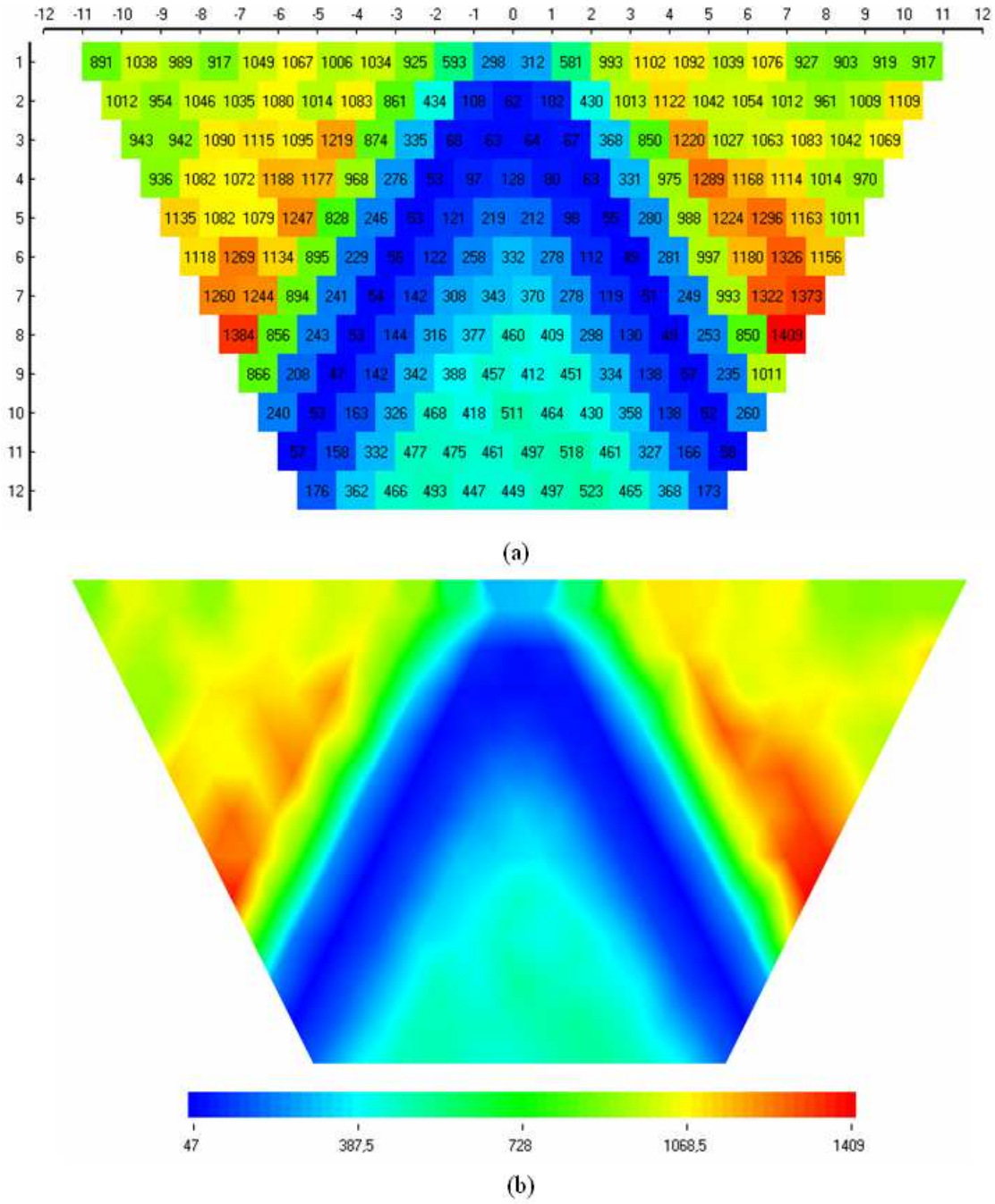


Figura 3.18: (a)Visualização da pseudo-seção da resistividade aparente editada, (b) Interpolação Otimizada da pseudo-seção da resistividade aparente editada.

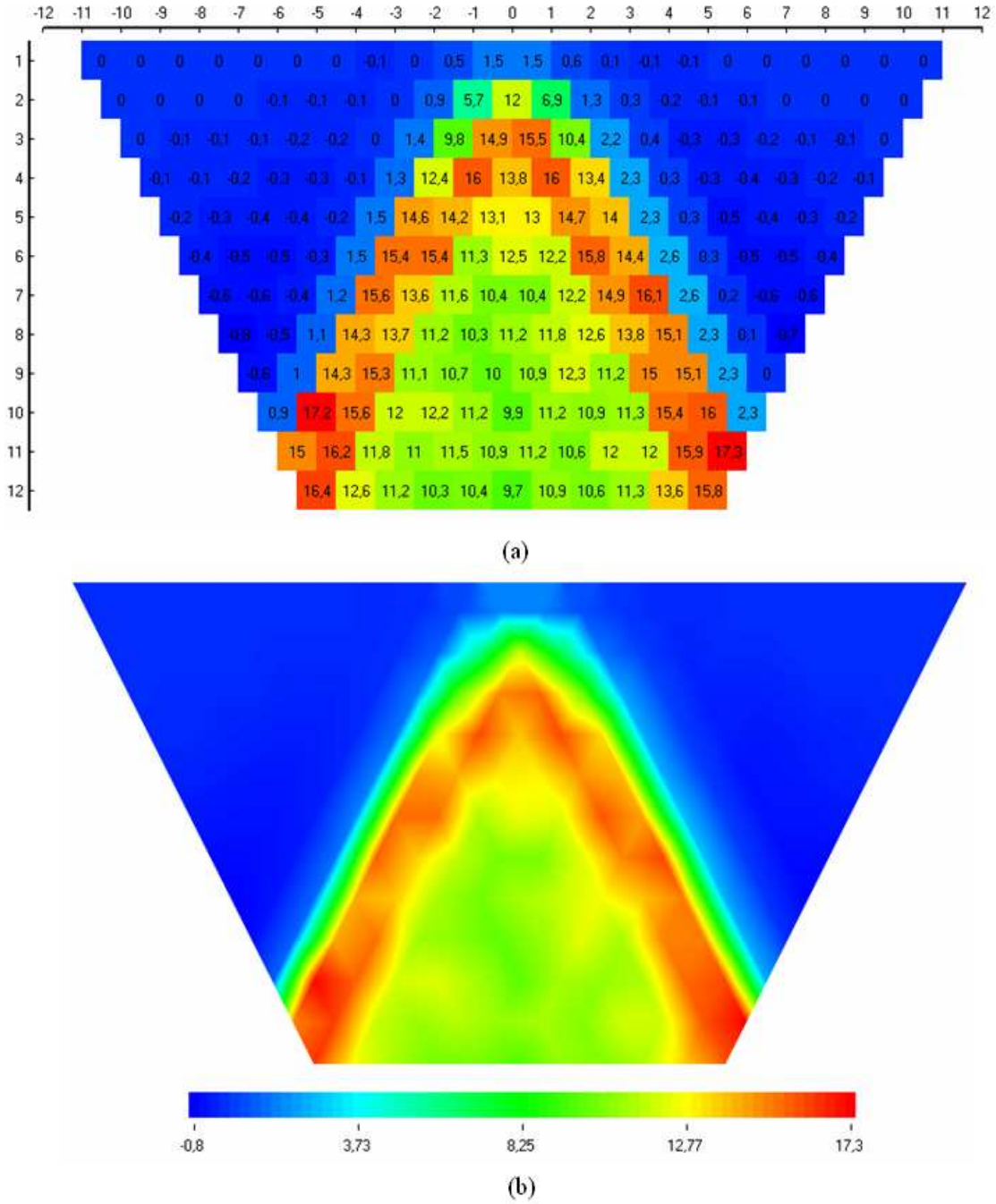


Figura 3.19: (a)Visualização da pseudo-seção do PFE editado, (b) Interpolação Otimizada da pseudo-seção do PFE editado.

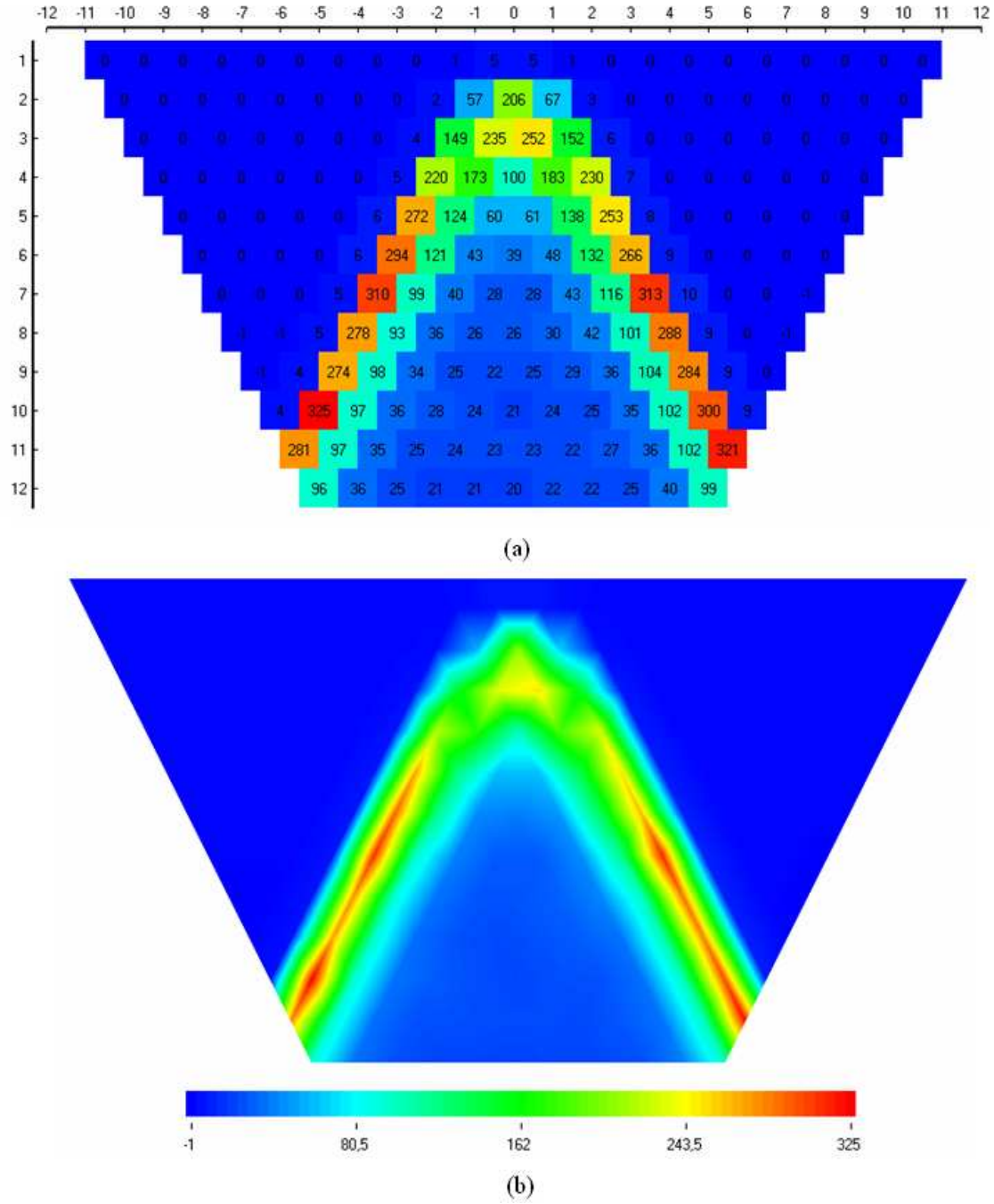


Figura 3.20: (a)Visualização da pseudo-seção do fator metálico editado, (b) Interpolação Otimizada da pseudo-seção do fator metálico editado.

Terminada a etapa de edição do *modelo geoelétrico* e *dados observados* de campo, a eGs está pronta para gerar a resposta do *modelo geoelétrico* para comparar com os *dados observados* de campo e verificar se o *modelo geoelétrico* é o mais adequado. Com um simples acionamento do botão *Executar*, a eGs 3.0 automaticamente fornece as pseudo-seções de resistividade aparente (Figura 3.21), PFE (Figura 3.22) e fator metálico (Figura 3.23), cujos valores são próximos aos mostrados nas Figuras 3.18, 3.19 e 3.20. Fazendo uma análise rápida, verifica-se que o modelo escolhido é o mais apropriado para fazer a interpretação dos *dados observados* obtidos no levantamento IP-Resistividade usado como exemplo.

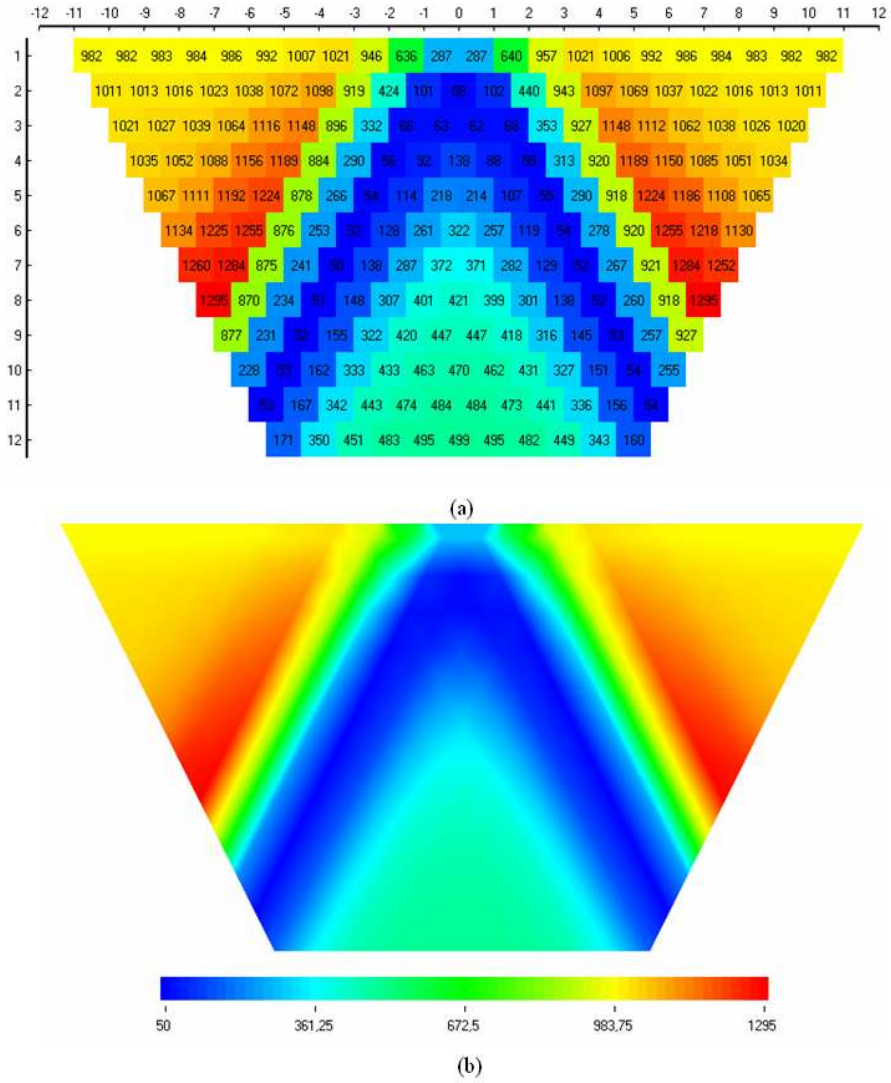


Figura 3.21: (a)Visualização do resultado da modelagem numérica da pseudo-seção da resistividade aparente, (b) Interpolação Otimizada do resultado da modelagem numérica da pseudo-seção da resistividade aparente.

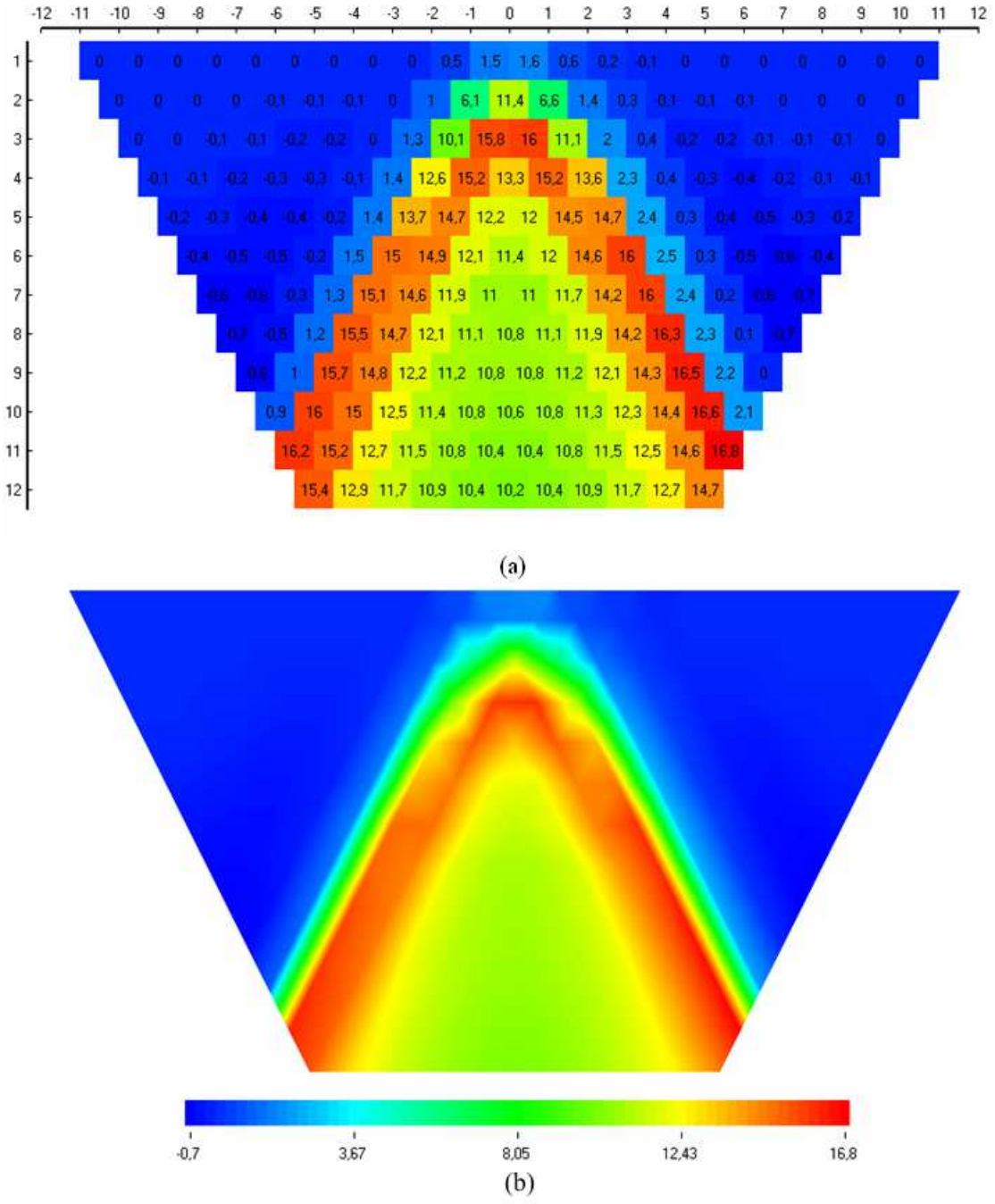


Figura 3.22: (a)Visualização do resultado da modelagem numérica da pseudo-seção do PFE, (b) Interpolação Otimizada do resultado da modelagem numérica da pseudo-seção do PFE.

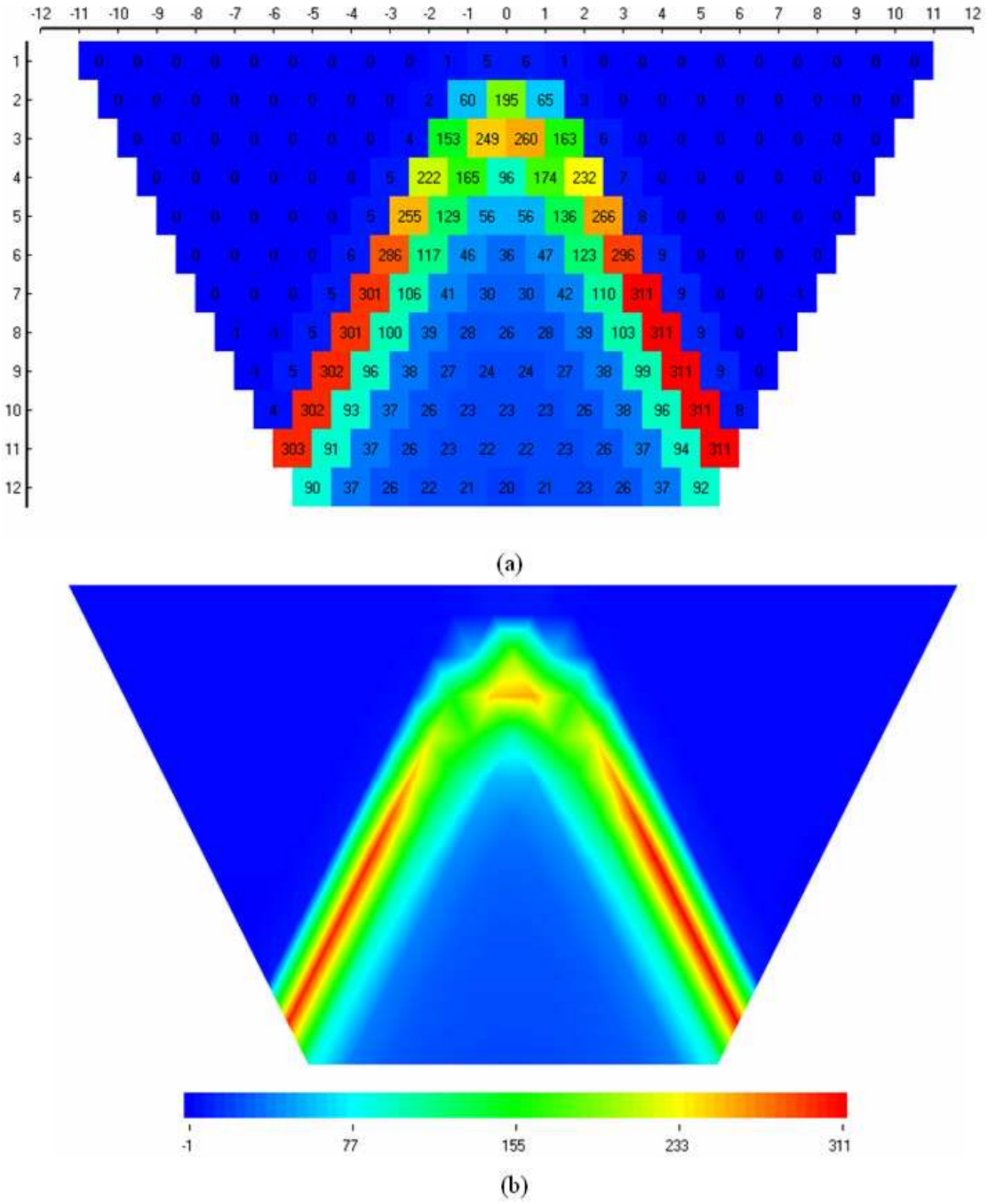


Figura 3.23: (a)Visualização do resultado da modelagem numérica da pseudo-seção do fator metálico, (b) Interpolação Otimizada da pseudo-seção do resultado da modelagem numérica do fator metálico.

4 - DESENVOLVIMENTO

Este capítulo se destina ao programador interessado em dar continuidade no projeto eGs. Àqueles que desejam apenas usar o aplicativo para modelar e interpretar dados, não é preciso saber os detalhes técnicos deste capítulo, podendo ir direto ao apêndice A, pois nele estão todas as informações de como manipular o software. A leitura deste capítulo requer algum conhecimento do ambiente de desenvolvimento Delphi e da linguagem *object pascal* (POO - pascal orientado a objetos).

4.1 - CONSTRUÇÃO.

Para se fazer a escolha de uma grande variedade de métodos geofísicos, para criar os *modelos geoelétricos*, usar os *dados observados*, executar a modelagem numérica e gerar as figuras para interpretação, decidiu-se desenvolver um software com multi-janelas que não dependesse da janela principal para as sub-janelas serem movimentadas na tela, ou seja, não fossem subordinadas às dimensões da janela pai, dando flexibilidade na manipulação das janelas. Isso era preciso, pois os métodos eletromagnéticos são utilizados em três categorias: uma, duas e três dimensões. Em cada dimensão possuem no mínimo cinco métodos com problema direto e inverso, sendo necessário ter o *modelo geoelétrico* e os *dados observados* de campo. Tendo que existir, também, as opções para a visualização de gráficos e das malhas em 2D e 3D.

Como o desenvolvimento da eGs foi divido, este capítulo descreverá a construção da janela "eGs 3.0", da janela "Métodos", da janela "Seleção", janelas "Dados Observados" e janelas "Modelo". As janelas utilizadas para visualizar os gráficos dos resultados da modelagem numérica e para malha 2D e 3D foi feita por Silva, R. E. C. (2005).

4.1.1 - Janela "eGs 3.0"

A maneira mais simples encontrada para o usuário ter acesso às opções e ferramentas existentes na eGs, foi criar um programa com um menu principal (Figura 4.1), que possui as opções padrões dos aplicativos desenvolvidos para *Windows*, onde é possível gerenciar todas as janelas existentes, as quais servem para a escolha da aplicação geofísica e para a inserção de dados já editados anteriormente.

Para a criação do menu e da barra de ferramentas, decidiu-se usar no formulário principal

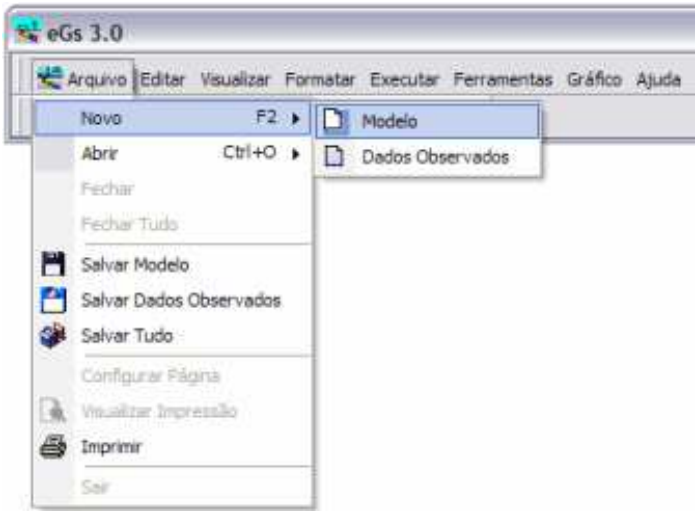


Figura 4.1: Menu *pulldown* da eGs 3.0

as componentes `Action-MainMenuBar` e `ActionToolBar` inseridas em um `ControlBar` com o `align` definido `alTop` para dar a aparência do *Windows XP*, pois o `MainMenu` usa a estrutura das versões antigas.

A escolha do `ControlBar` foi com a finalidade de arrastar o menu e a barra de ferramentas dentro da janela ou fora dela. Para isso, foram alteradas as seguintes propriedades no *Object Inspector*:

- `AutoDock := True` : Essa propriedade serve para aceitar o menu e a barra de ferramentas, quando elas forem arrastadas para fora da janela principal (Figura 3.5). Caso a atribuição fosse `False`, o menu e a barra não seriam aceitos pela janela "eGs 3.0";
- `AutoDrag := True` : Essa propriedade serve para arrastar o menu e a barra de ferramentas;
- `AutoSize := True` : Essa propriedade tem a função de deixar a altura sempre igual à soma das componentes inseridas no `ControlBar`, ou seja, se `ActionMainMenuBar` e `ActionToolBar` estiverem incluídos. O que se torna o equivalente aos seguintes comandos:

```
ControlBar.Height:= ActionMainMenuBar1.Height + ActionToolBar1.Height;
```

Caso contrário:

```
ControlBar.Height:= ActionMainMenuBar1.Height;
```

Ou:

```
ControlBar.Height:= ActionToolBar1.Height;
```

O único método do `ControlBar` utilizado foi o `OnResize`, onde seu código é:

```
procedure TForm00_MenuPrincipal.ControlBarResize(Sender:TObject);
begin
```

Início do Comando 4.1

```
 ClientHeight:=ControlBar.Height;
```

Fim do Comando 4.1

```
end;
```

O Comando 4.1 tem por finalidade deixar sempre a janela principal, que é o menu, da altura do `ControlBar`. Para a opção de visualizar o menu e a barra de ferramentas, associou-se a componente `PopupMenu`. Para atualizar as marcações foi escrito no evento `OnPopup` o Comando 4.2.

```
procedure TForm00_MenuPrincipal.PopupMenu1Popup(Sender: TObject);
var
  I: Integer;
begin
```

Início do Comando 4.2

```
// atualiza as marcacoes do PopupMenu
for I := 0 to PopupMenu1.Items.Count - 1 do
 PopupMenu1.Items [I].Checked := TControl (PopupMenu1.Items[I].Tag).Visible;
```

Fim do Comando 4.2

```
end;
```

Para incluir os botões no menu, usou-se uma componente muito útil, embora pouco empregada por desenvolvedores, chamada de `ActionManager`. Ela tem a capacidade de criar procedimentos, chamados de `Action`, um objeto do tipo `TAction`, que são usados para centralizar os comandos de resposta para os usuários e com intuito de representar os elementos utilizados da interface na aplicação.

Para gerenciar o item *Novo→Modelo*, *Novo→Dados Observados*, *Salvar* e *Abrir*, fez-se primeiramente `Actions`. Para isso, bastou ir ao `ActionManager` e em sua a janela "Actions" existe o botão "New Actions(Ins)", o qual insere novas ações. Para codificar procedimentos, clicou-se duas vezes e o Delphi produziu um espaço no código fonte para escrever os comandos.

Depois, no *Object TreeView* do Delphi, onde é possível visualizar `ActionsBars` o qual serviu para ligar as ações com `ActionMainMenuBar` para criar o menu e `ActionToolBar` para criar a barra de ferramentas.

Para associar `Actions` do `ActionManager` ao `ActionMainMenuBar`, clicou-se com o botão direito do mouse no `ActionsBars` do tipo `ActionManager.ActionsBars`, surgindo `ActionManager.ActionsBars[0]` para criar o menu, e `ActionManager.ActionsBars[1]` para criar a barra de ferramentas. Todos os objetos `TActionBarItem` possuem em sua árvore um subitem chamado *Item* da classe `TActionClientItem`, assim em `ActionManager.ActionsBars[0]` foi adicionado um `TActionClientItem[i]` (onde i = 0, 2, 2, ..., 7) para criar as opções *Arquivo*, *Editar*, *Visualizar*, *Ajuda* (Figura 3.4). Por exemplo, o elemento de índice 0 é a opção *Arquivo* em tempo de execução. Para criar as sub-opções do menu, por exemplo, do item *Arquivo*, foi adicionado `TActionClientItem[j]` (onde j = 1, 2, ..., 10) no `TActionClientItem[0]`. Todos os `TActionClientItem` possuem a propriedade `Action` e nela estão listados os `Actions` feitos no `ActionManager`, podendo portanto fazer a ligação.

Para vincular as ações ao `ActionMainMenuBar` era preciso apenas usar a propriedade `ActionsBar` do `ActionManager.ActionsBars[0]`, mas preferiu-se escrever o comando no evento `OnCreate` do formulário `Form00_MenuPrincipal`, pois foi o único modo achado para que a altura do `ActionMainMenuBar` não ficasse grande em tempo de execução, o que não ocorre quando é atribuído o valor da propriedade através do *Object Inspector*. Veja o Comando 4.3.

```
procedure TForm00_MenuPrincipal.FormCreate(Sender: TObject);
var
  I,Tempo: Integer;
  mItem: TMenuItem;
begin
//***************
Modelo2D:=False;
//***************
NodeModelo  :=0;
NodeDadosObs:=0;
Form00_MenuPrincipal.Hide;
//cmd p/ o splash permanecer por um tempo\\
Tempo:= GetTickCount div 1000;
```

```
while (GetTickCount div 1000) <(Tempo+4) do;
//cmd p/ Definir o Tamanho do Menu Principal\\
With Screen do
 begin
  Left:=WorkAreaLeft;
  Top:=WorkAreaTop;
  Form00_MenuPrincipal.Width:=WorkAreaWidth;
 end;
ActionMainMenuBar1.Width:=ClientWidth;
```

Início do Comando 4.3

```
//cmd p/ Ligar as acoes com as componentes\\
{Obs:foi o unico modo de achei para a altura do
actionMainMenuBar nao ficar grande.}
with ActionManager1.ActionBars do
 begin
  ActionBars[0].ActionBar:=ActionMainMenuBar1;
  ActionBars[1].ActionBar:=ActionToolBar1;
 end;
```

Fim do Comando 4.3

```
{ComboFont.Items := Screen.Fonts;}
//Coloca as componentes no PopupMenu \\
for I := 0 to ControlBar.ControlCount - 1 do//<----|
 begin                                            //|
  mItem := TMenuItem.Create (Self);               //|
  mItem.Caption := ControlBar.Controls [I].Name; //|
  mItem.Tag := Integer (ControlBar.Controls [I]);//|
  mItem.OnClick := BarMenuClick;                  //|
  PopupMenu1.Items.Add (mItem);                   //|
 end;//<-----------------------------------------|
```

Na opção *Arquivo*, criou-se um botão *Novo*, que possui dois sub-itens para abrir a janela, uma "Modelo" e a outra "Dados Observados", onde o procedimento para mostrar uma janela nova para edição seguiu o raciocínio do Fluxograma mostrado na Figura 2.2. Caso o usuário

tente passar por alguma etapa, por exemplo, queira editar um *modelo geoelétrico* novo sem escolher um método, será mostrada uma janela informativa (Figura 3.9.(a)).

Caso o usuário escolha o método e não indique o tipo de problema, ocorre o mostrado na Figura 3.9.(b).

Depois de editado o arquivo do *modelo geoelétrico* ou dos *dados observados*, ele pode ser salvo usando a componente `savedialog`, a qual abre uma janela modal (Figura 3.6) para escolher o local para salvar o arquivo.

Para abrir um *modelo geoelétrico* ou um *dado observado* salvo anteriormente, usa-se o mesmo raciocínio utilizado para um arquivo novo. Porém, o processo é inverso devido ao arquivo já existir. Sendo assim, usou-se a componente do Delphi `opendialog`, que tem a mesma estrutura da `savedialog`, mas com a diferença de abrir o arquivo.

Tanto o `opendialog` quanto `savedialog`, após executadas essas componentes no aplicativo, possuem a propriedade `Filename`, que tem a finalidade de armazenar o diretório do arquivo.

Cada arquivo possui uma formatação específica que depende do método, devido as dlls serem exigentes nas leituras dos dados. Porém, pensou-se em uma estrutura após edição dos arquivos de modo que não fosse demasiadamente diferente, ou seja, seguisse uma lógica. Assim esse padrão é:

- Extensão para os Arquivos: Pensou-se em extensões para diferenciar os arquivos dos *modelos geoelétricos* dos *dados observados*, que são: *.MGE e *.OBS.
- Chave no Início dos Arquivos: Para diferenciar os métodos e a dimensão que está sendo usada, decidiu-se usar uma "chave" na primeira linha do arquivo para facilitar na identificação. Por exemplo, caso o usuário salve um arquivo em que a dimensão escolhida foi a 1D e o método foi o magnetotelúrico, o aplicativo irá editar MT1D antes de salvar os dados editados na janela.

4.1.2 - Janela "Métodos"

Após a criação do menu, houve a necessidade de uma interface que facilitasse na escolha da dimensão, dos métodos e do problema direto ou inverso. Para isso, criou-se uma janela com uma árvore semelhante a do *Windows Explorer* devido a grande quantidade de técnicas na área de métodos geofísicos elétricos eletromagnéticos, para mostrar os métodos implementados na dimensão desejada. Veja a Figura 3.7.

Para criar a janela "Métodos" foram utilizados três `RadioButton` dentro de um `RadioGroup` para permitir a escolha das dimensões e a troca da árvore de métodos que foram construídas usando a componente `TreeView` do Delphi, cuja sua funcionalidade é baseada na seleção da linha, ou seja, cada linha escolhida possui um valor inteiro na propriedade `Selected.AbsoluteIndex`, sendo possível assim saber qual método o usuário deseja usar sabendo que os itens são numerados de cima para baixo e iniciando do zero. Logo, fez-se uma função chamada de `ItemdaArvore` na unit `UnitArvore` que retorna apenas com o número da linha. Veja o Comando 4.4.

Início do Comando 4.4

```
function TForm00_Arvore.ItemdaArvore:longint;
begin
if RadioButton1D.Checked then
 Result:=TreeView1.Selected.AbsoluteIndex
else if RadioButton2D.Checked then
 Result :=TreeView2.Selected.AbsoluteIndex
     else
      Result := TreeView3.Selected.AbsoluteIndex;
end;
```

Fim do Comando 4.4

4.1.3 - Janela "Seleção"

O problema seguinte era uma outra opção para o usuário escolher o tipo de arquivo que deseja editar ou abrir que não fosse no menu. Assim criou-se uma janela para mostrar os *modelos geoelétricos* ou *dados observados* e executar a modelagem numérica do método escolhido (Figura 3.8).

Os botões desse formulário usam os `RadioButton` do formulário `Form00_Arvore` e a função `ItemdaArvore` mostrada na seção anterior, para saber que dimensão e método foram escolhidos para criar o formulário correto. O Comando 4.5 foi desenvolvido para novos formulários dos *modelos geoelétricos* em uma dimensão, mas o raciocínio é o mesmo para novos formulários dos *dados observados* e *modelos geoelétricos*s em outras dimensões.

```
procedure TForm00_MenuPrincipal.NovoModeloExecute(Sender:TObject);
```

```
const
  Mod1='Modelo';
begin
SalvarModelo.Enabled:=True;
if SalvarModelo.Enabled andSalvarDadosObs.Enabled then
 SalvarTudo.Enabled:=True;
```

Início do Comando 4.5

```
//Abre formularios de metodos em 1D
if Form00_Arvore.RadioButton1D.Checked then
 begin
 //**************
  Modelo2D:=False;
 //**************
  Case Form00_Arvore.ItemdaArvore of
   //-----Magnetotelurico Isotropico--------------\\
   //-----Eletrojato Equatorial Linha Infinita----\\
   //-----Eletrojato Equatorial 1 Gaussiana-------\\
   //-----Eletrojato Equatorial 3 Gaussiana-------\\
   //-----Linha de Corrente Infinita--------------\\
   //-----Bobina Circular-------------------------\\
   //-----Dipolo Magnetico Vertical---------------\\
   2,3,11,12, 14,15, 17,18, 21,22, 28,29, 35,36:
    begin
     NodeModelo:=NodeModelo+1;
     Form01_MTModelo:= TForm01_MTModelo.Create(Application);
     Form01_MTModelo.Show;
     Form00_Selecao.Edit1.Text:= Mod1+IntToStr(NodeModelo);
    end;
   //-----Sondagem Elétrica Vertical-------\\
   56,57,59,60,62,63:
    begin
     NodeModelo:=NodeModelo+1;
     Form1DSevMod:= TForm1DSevMod.Create(Application);
     Form1DSevMod.Show;
```

```
    Form00_Selecao.Edit1.Text:= Mod1+IntToStr(NodeModelo);
   end;
  //-----Magnetotelurico Marinho-------\\
  8,9:
   begin
    NodeModelo:=NodeModelo+1;
    FormMcsem1dMge:= TFormMcsem1dMge.Create(Application);
    FormMcsem1dMge.Show;
    Form00_Selecao.Edit1.Text:= Mod1+IntToStr(NodeModelo);
   end;
 end;
```

Fim do Comando 4.5

```
 Case Form00_Arvore.ItemdaArvore of
  0,19,26,54:MessageDlg('Selecione um Metodo',mtInformation,[mbOK],0);
  1,4,7,20,23,27,30,55,58,61:MessageDlg('Selecione ou Direto, ou Inverso',
  mtInformation,[mbOK],0);
 end;
end;
//Abre formularios de metodos em 2D
if Form00_Arvore.RadioButton2D.Checked then
 begin
 //**************
  Modelo2D:=True;
  ItemDaArvore2D:=Form00_Arvore.ItemdaArvore;
  ItemDaArvore3D:=-1;
 //**************

  Case Form00_Arvore.ItemdaArvore of
   //-----Magnetotelurico------------------------\\
   1://Modo TETM
    begin
     //NodeModelo:=NodeModelo+1;

     ChaveSalvar:='TETMMT2D';
```

```
     Form4_Malha3D.Show;
     Form00_Selecao.Edit1.Text:= Mod1+IntToStr(NodeModelo);
    end;
   //-----Very Low frequency (VLF)----------------\\
   4:
    begin
     //NodeModelo:=NodeModelo+1;

     ChaveSalvar:='VLF2D';//Rodrigo (05.12.03)
     Form4_Malha3D.Show;
     Form00_Selecao.Edit1.Text:= Mod1+IntToStr(NodeModelo);
    end;
   7:
    begin
     //NodeModelo:=NodeModelo+1;

     ChaveSalvar:='RESIP2D';//Rodrigo (05.12.03)
     Form4_Malha3D.Show;
     Form00_Selecao.Edit1.Text:= Mod1+IntToStr(NodeModelo);
    end;
  end;//fim do case
 end;
//Abre formularios de metodos em 3D
if Form00_Arvore.RadioButton3D.Checked then
 begin
 //**************
  Modelo2D:=False;
  ItemDaArvore2D:=-1;
  ItemDaArvore3D:=Form00_Arvore.ItemdaArvore;
 //**************
  Case Form00_Arvore.ItemdaArvore of
   0:
    begin
     Form4_Malha3D.show;{Inserir Comando}
```

```
    end;
  end;
 end;
end;
```

O procedimento utilizou o laço `case`, sendo que as condições são os números das linhas na componente `Treeview` . Decidiu-se criar os formulários com o comando:

```
Form01_MTModelo:= TForm01_MTModelo.Create(Application)
```

Pois assim é possível abrir várias janelas para serem editadas ao mesmo tempo. Como é possível observar o Comando 4.5 que existem vários números (métodos) para o mesmo formulário, isto ocorre porque era preciso padronizar as janelas "Modelo", de modo que o usuário precisasse saber como manipular apenas uma janela, pois se cada método EM possuísse uma entrada de *modelos geoelétricos* diferente, o usuário que iria usar a eGs nesse formato talvez achasse mais fácil aprender uma linguagem de programação para manipular as dlls do que aprender a usar as janelas método por método.

O botão *Abrir* possui a mesma lógica do botão *Novo*, ou seja, quem induz o programa a mostrar o formulário correto é o número inteiro `Form00_Arvore.ItemdaArvore`. O detalhe é que ocorreu um problema quando se decidiu criar os *modelos geoelétricos*, abri-los com o `OpenDialog` logo em seguida e depois mostrar a janela com os arquivos. O erro nesse caso é que o primeiro valor existente no arquivo não é lido, alterando assim a seqüência de leitura dos dados, e conseqüentemente, inserindo valores em campos diferentes da janela. A resolução do problema foi simples: bastou mostrar a janela atrávés do procedimento `Show` do formulário antes de executar a janela modal de abertura de arquivos.

O botão *executar* também pode ser usado no menu. A sua função é simplesmente ativar a dll do método escolhido na árvore de métodos EM. Como é possível na eGs fazer gráficos dos arquivos criados pela dll, pensou-se em fazer duas opções para o usuário, sendo que a primeira executa o problema direto ou inverso e faz as curvas com os arquivos de saída para a interpretação geofísica, e a segunda apenas utiliza o programa FORTRAN e salva os dados no mesmo diretório do *modelos geoelétricos*.

4.1.4 - Janelas "Modelo"

É extremamente necessário que na opção 1D a eGs tenha janelas que representem os *modelos geoelétricos* teóricos em camadas estratificadas com suas respectivas propriedades físicas

como resistividade e permeabilidade magnética; e geométricas como a espessura. Logo, nessa fase de construção o programador teve como objetivo a criação de formulários padronizados, onde a mudança de método elétrico e eletromagnético não alterasse na estrutura de edição de dados, de modo que facilitasse na implementação dos métodos e, o mais importante, o usuário não precisasse fazer um *modelo geoelétrico* diferente para cada aplicação geofísica.

Para criar as janelas, primeiramente, foi necessário saber como deveriam ser os arquivos dos *modelos geoelétricos* para que as dlls os utilizassem sem erro de leitura. Os dados devem representar o meio estratificado em camadas, com suas respectivas propriedades elétricas (Figura 4.2(a)), portanto, chegou-se à estrutura mostrada na Figura 4.2(b), sendo que a primeira linha é a chave para a eGs identificar o método e a dimensão escolhida. Na segunda linha, estão os valores do número de camadas e o percentual de ruído nos resultados finais. E os valores seguintes, formatados em colunas, são as resistividades, permeabilidades e espessuras. Como é possível observar na Figura 4.2(a) e (b), como exemplo, que a última camada, chamada de substrato, não possui espessura.

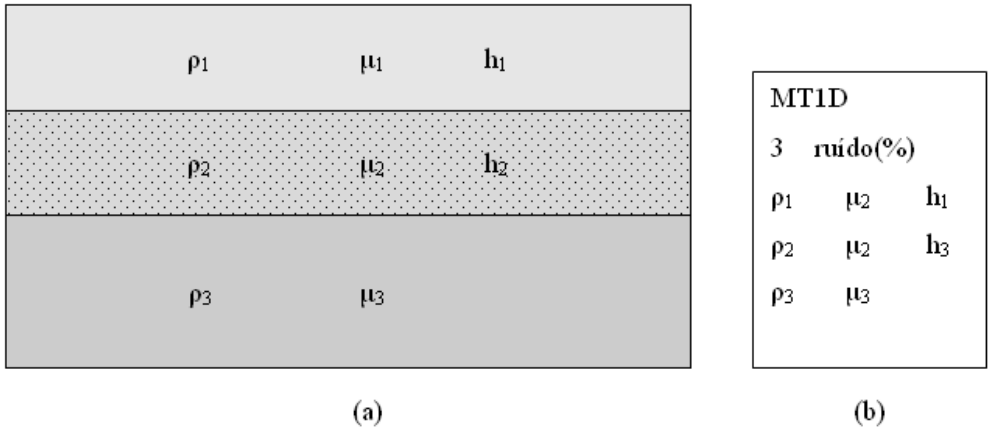


Figura 4.2: (a) *modelos geoelétricos* de um meio estratificado (b) Formato do arquivo para uma dimensão.

Para as dlls, os dados devem ser salvos em arquivos de texto com extensão *.MGE, ou seja, todos os valores editados nos *modelos geoelétricos* são um conjunto de caracteres e não números inteiros ou reais, pois se fosse definido como um arquivo do tipo ponto flutuante, por exemplo, os dados seriam salvos apenas em uma linha e criptografados, não podendo ser usados. Isso dificultou na estruturação do código para salvar os arquivos na formatação correta, mas facilitou no armazenamento, pois arquivos do tipo texto não ocupam muito espaço na memória, enquanto que em outros formatos teriam problemas tanto no armazenamento quanto na leitura, pois o funcionamento dessas duas ações seria mais lento.

Os formulários para os *modelos geoelétricos* construídos para uma dimensão seguem, basicamente, a estruturação mostrada na Figura 3.10, onde a primeira caixa de edição é para inserir o número de camadas através do teclado ou pela componente *updown* que pode utilizar o mouse, dando assim uma flexibilidade na edição dos dados. Os valores do número de camadas servem para atualizar, automaticamente, a grade, onde a primeira coluna é reservada para as camadas, a segunda para a resistividade, a terceira para permeabilidade, com exceção das sondagens elétricas verticais Schlumberger, Wenner e Dipolo-Dipolo, que não usam a permeabilidade magnética relativa, e a última coluna é destinada para edição das espessuras.

Na fase de desenvolvimento surgiu um problema: percebeu-se que a caixa de edição para o número de camadas não deveria ficar vazia, ou seja, sempre teria que existir um valor contido, pois quando excluídos todos os números daria sempre um erro em tempo de execução. Para resolver isso, criou-se um laço condicional no evento `Onchange` do `edit`, de modo que o programa não faz nenhuma ação quando a caixa está vazia, mas apenas isso não foi necessário para que o erro deixasse de ocorrer. Usou-se então a função `Val(Edit1.Text,No,Cod))`, onde o primeiro parâmetro (`Edit1.Text`) deve ser uma variável do tipo `string` que se deseja converter para um número inteiro, o segundo (`No`) uma variável definida como `Integer` para receber a conversão e o terceiro (`Cod`) é a variável a qual verifica se a `No` pode ou não ser um número inteiro, retornando `Cod := 0` se possível e `Cod <>0` caso seja impossível (veja o Comando 4.6).

Início do Comando 4.6

```
procedure TForm01_MTModelo.Edit1Change(Sender: TObject); Var
  No, Cod:Integer;
begin
if
 Edit1.Text<>'' then
  begin
   Val(Edit1.Text,No,Cod);
   if (Cod <>0) or (No=0) then
    begin
     MessageDlg('O Numero de Camadas deve ser entre 1 e 100',
     mtInformation,[mbOK],0);
     Edit1.Text:='';
```

```
     Edit1.SetFocus;
    end
   else if No<=100  then
    begin
     StrGrid1.RowCount:=No+1;
     StrGrid1.Repaint;
     StsBar1.Panels[0].Text:='Camada: '+ Edit1.Text+';
     '+' Ruido: '+Edit2.Text+'%';
    end
       else
        begin
         MessageDlg('O Numero de Camadas deve ser entre 1 e 100',
         mtInformation,[mbOK],0);
         Edit1.SetFocus;
        end;
 end;
end;
```

Fim do Comando 4.6

Na última camada não existe espessura, portanto para evitar que o usuário não edite esse dado desnecessariamente foi feita uma condição (Comando 4.7), onde desabilita a célula da terceira coluna e última linha da componente `StringGrid` para edição e informa ao usuário.

```
procedure TForm01_MTModelo.StrGrid1KeyPress(Sender: TObject; var
Key: Char);
begin
if not (Key in ['0'..'9','-','+','.',DecimalSeparator,Chr(8)]) then Key:= #0
else Modificado:=true;

if (Key in ['.']) then key:=',';
```

Início do Comando 4.7

```
if (StrGrid1.Row = StrToInt(Edit1.Text)) and (StrGrid1.Col = 3) then
 begin
  key:= #0;
```

```
  MessageDlg('Nao existe a ultima Espessura!',mtInformation,[mbOK],0);
 end;
```

Fim do Comando 4.7

```
end;
```

No Comando 4.7 a variável `key` é do tipo `Char`, que indica qual tecla está sendo pressionada no teclado, quando atribui a ela o valor `#0` desativa a ação do periférico de edição de texto. A última linha do laço é a função `MessageDlg('texto' ,mtInformation,[mbOK],0)` a qual mostra uma janela do tipo `mtInformation`, com apenas um botão do tipo `mbOK` e com um texto inserido (veja Figura 3.12).

4.1.5 - Janelas "Dados Observados"

É indiscutível que seria necessário dispor janelas para edição, alteração e armazenamento das atividades de levantamento, pois o objeto principal da eGs é exatamente compreender e ter domínio dos valores obtidos através dos levantamentos eletromagnéticos. Ao contrário do *modelos geoelétricos*, nos *dados observados* não existe uma estrutura uniformizada das propriedades físicas entre os métodos, devido os valores não serem teóricos e sim, adquiridos na natureza pela necessidade da aplicação geofísica. Mas entre as técnicas a singularidade desaparece quando muda a dimensão, podendo fazer um formulário para uma dimensão (1D) e ser aproveitada em 2D e 3D. Assim, foram construídas janelas para cada um dos métodos com a intenção de ler e salvar arquivos do tipo texto estandardizados na extensão (*.OBS) e na formatação interna dos dados.

Os formulários para os *dados observados* segue duas estruturas:

1. **Na forma de tabelas:** Há uma caixa de edição para inserir o número de linhas que a tabela deve possuir, as colunas para incluir os valores específicos de cada método, como é mostrado na Figura 3.13(a) e (b);

2. **Na forma de pseudo-seções:** As pseudo-seções são perfis de levantamento na superfície, como mostra a Figura 3.14;

A estratégia utilizada para a construção dos formulários foi iniciar com um método cuja edição fosse a mais simples até a mais complicada para a implementação. Logo, a sondagem elétrica vertical é a que possui a forma mais simplificada para leitura e armazenamento dos dados, pois deve existir apenas o número de observações (n), os arranjos dos eletrodos (AB/2) e a resistividade aparente (ρ_a). Observando a Figura 4.3, verifica-se na primeira linha

a palavra SEV1D, a qual é a chave para o programa identificar o método e a dimensão em que o arquivo pode ser utilizado. Apesar de não ser necessário diferenciar a dimensão, decidiu-se distinguir para o caso dos dados dos novos métodos que serão incluídos não possuírem semelhança entre as dimensões.

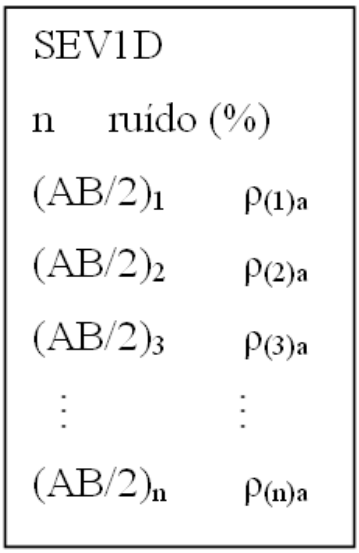


Figura 4.3: Formato do arquivo para uma dimensão.

Para o usuário editar e visualizar arquivos salvos dos *dados observados* de sondagens elétricas verticais, foi desenvolvido o formulário `Form1DSevDadosObs` para a geração da janela mostrada na Figura 3.13.(a).

A estrutura mais complicada na forma de tabela é a janela do eletrojato equatorial (Figura 3.13.(b)), onde seu objetivo é representar um levantamento, ilustrado na Figura 4.4 de k observações em posições colineares D_j ($1 \leq j \leq k$) da estação em um plano α, sendo que cada obtenção dos valores dos períodos T_i, resistividade aparente $\rho_{(i)a}$ e fase ϕ_i, onde $1 \leq i \leq n$, foram adquiridos em n períodos de uma corrente de linha I_0 contida em um plano β a uma altura h_0 de α, com desvio padrão σ e contra-eletrojatos A_1 e A_2 (Mota, 1991; Carrasquilla, 1993 e Silva, M. W., 2005).

A segunda estrutura de formulário, foi desenvolvido para o método geofísico em 2D da polarização induzida e resistividade (RES-IP), pois os valores inseridos na janela devem estar organizados na forma de pseudo-seções de IP (Guimarães, 1983). Assim é preciso que o número de linhas seja a metade do número de colunas menos um, com as caixas de edição dispostas em forma similar a um trapézio, como mostra a Figura 3.14. Em apenas uma janela é possível editar três tipos de propriedades físicas: resistividade aparente (ρ_a), percentual do

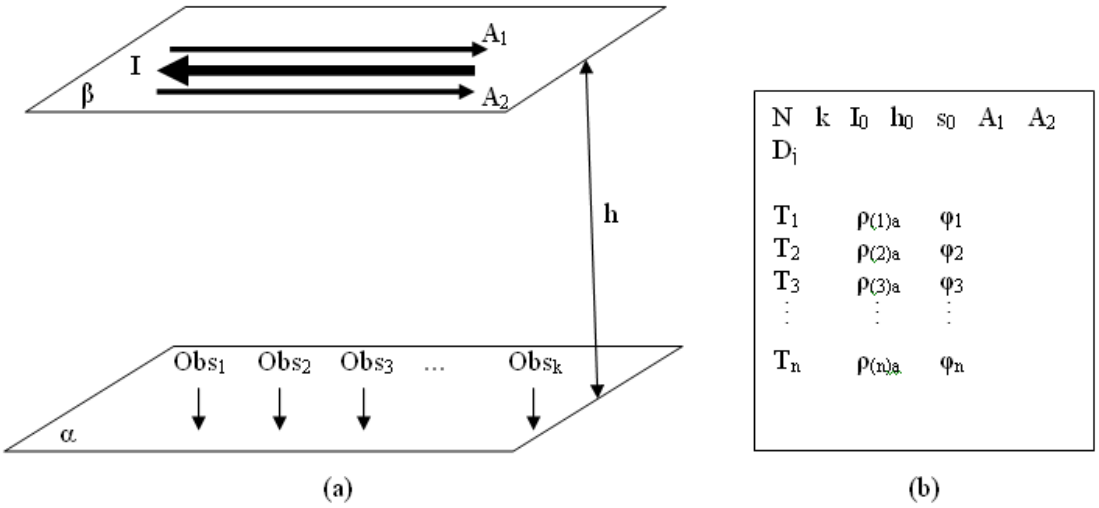


Figura 4.4: (a) Representação geométrica do eletrojato equatorial 3 gaussiana (b) Formato do arquivo para eletrojato equatorial 3 gaussiana.

efeito da freqüência (PFE - Percentual Frequency Effect) e o fator metálico (MF - Metal Factor).

A característica mais importante desenvolvido para o IP e Resistividade foi criar as caixas de edição de valores, as linhas e as numerações que determinam os níveis e os eletrodos da janela de forma inteligente, de modo que o programa saiba quando o número de eletrodos ou de níveis não satisfaz a condição da equação 4.1 e acrescente, ou diminuía, o número de eletrodos ou de níveis para continuar atendendo a Equação 4.1.

$$N^o\ de\ Niveis = \frac{N^o\ de\ Eletrodos}{2} - 1 \tag{4.1}$$

Portanto, para iniciar os comandos para a criação das componentes no formulário para IP e Resistividade, primeiramente foram declaradas as variáveis `NodeEditAntes`, `NodeNiveisAntes`, `ShapeEletrodos`, `ShapeNiveis`, `MatrizLabelEletrodo` e `MatrizLabelNiveis` na seção `public` da unit `Unit2DResIPDadosObs`. Os números inteiros `NodeEditAntes` e `NodeNiveisAntes` serão sempre a quantidade editada no `EditEletrodos` menos três e no `EditNiveis`, respectivamente, anterior às suas mudanças. `ShapeEletrodos` e `ShapeNiveis` são matrizes do tipo `TShape` de uma dimensão onde são guardadas as linhas pretas verticais para a representação da posição dos `eletrodos` e horizontais para indicar os níveis. A numeração vertical e horizontal foram feitas usando `labels`, os quais são elementos das matrizes `MatrizLabelEletrodo` para numerar os eletrodos e `MatrizLabelNiveis` para o nivelamento. O passo seguinte foi

criar o edit00 no evento OnCreate do Form2DResIPDadosObs com o Comando 4.8.

```
procedure TForm2DResIPDadosObs.FormCreate(Sender: TObject);
var
  i:integer;
begin
//*******************************
 Nmalhas:=3;
 LeftGrid:=0;      TopGrid:=0;
 LeftGridAnt:=0;  TopGridAnt:=0;
 LadoQ:=30;
 MaxLadoQInterpol:=8;

 ImageInterpolaIP:=TImage.Create(Self);
 ImageInterpolaIP.Parent:=ScrollBoxInterpolar;
 ImageInterpolaIP.Left:=0;   ImageInterpolaIP.Top:=0;
 ImageInterpolaIP.Width:=50; ImageInterpolaIP.Height:=50;
//*******************************

//Armazena_ResAp:=True; Armazena_PFE:=False; Armazena_MF:=False;
Armazena_ResAp_PFE_MF[0]:=True;  //Resistividade Aparente
Armazena_ResAp_PFE_MF[1]:=False; //PFE
Armazena_ResAp_PFE_MF[2]:=False; //Fator Met{\'a}lico;

NodeEditAntes:=1; NodeNiveisAntes:=1;
```

Início do Comando 4.8

```
//cria o Edit0
SetLength(MatrizEdits,1,1);
MatrizEdits[0,0] := TEdit.Create(Self);
MatrizEdits[0,0].Parent := ScrollBox1;
MatrizEdits[0,0].Width := 60;
MatrizEdits[0,0].Height := 21;
MatrizEdits[0,0].Left := Round(Shape1.Left+MatrizEdits[0,0].Width+2.5);
MatrizEdits[0,0].Top := Shape2.Top;
```

```
MatrizEdits[0,0].Name := 'Edit00';
MatrizEdits[0,0].Text := '';
```

Fim do Comando 4.8

Início do Comando 4.9

```
SetLength(ShapeNiveis, StrToInt(EditNiveis.Text));
SetLength(MatrizLabelNiveis, StrToInt(EditNiveis.Text));
SetLength(ShapeEletrodos, StrToInt(EditEletrodos.Text));
SetLength(MatrizLabelEletrodo, StrToInt(EditEletrodos.Text)); for
i:=0 to StrToInt(EditEletrodos.Text)-1 do
 begin
  //cria os Shapes dos Eletrodos
  ShapeEletrodos[i] := TShape.Create(Self);
  ShapeEletrodos[i].Parent := ScrollBox1;
  if i>1 then
   ShapeEletrodos[i].Left :=
   Round(ShapeEletrodos[i-1].Left+MatrizEdits[0,0].Width+5)
  else
   ShapeEletrodos[i].Left := Round(Shape1.Left+i*(MatrizEdits[0,0].Width));
  ShapeEletrodos[i].Top := Shape1.top-5;
  ShapeEletrodos[i].Width := 1;
  ShapeEletrodos[i].Height := MatrizEdits[0,0].top - Shape1.top;
  ShapeEletrodos[i].Name := 'ShapeEletrodos' + IntToStr(i);
  //cria os labels dos Eletrodos
  MatrizLabelEletrodo[i] := TLabel.Create(Self);
  MatrizLabelEletrodo[i].Parent := ScrollBox1;
  MatrizLabelEletrodo[i].Width :=
  MatrizLabelEletrodo[i].Canvas.TextWidth(IntToStr(i));
  MatrizLabelEletrodo[i].Left :=
  ShapeEletrodos[i].Left-MatrizLabelEletrodo[i].Width;
  MatrizLabelEletrodo[i].Top := ShapeEletrodos[i].Top -20;
  MatrizLabelEletrodo[i].Caption:=IntToStr(i+1);
  if i=0 then
   begin
    ShapeEletrodos[i].Left := Shape1.Left;
    //cria os Shapes dos N{\'\i}veis
```

```
    ShapeNiveis[i] := TShape.Create(Self);
    ShapeNiveis[i].Parent := ScrollBox1;
    ShapeNiveis[i].Left := Shape2.Left -5;
    ShapeNiveis[i].Top :=
    MatrizEdits[0,0].top+(MatrizEdits[0,0].Height div 2);
    ShapeNiveis[i].Width :=
    (MatrizEdits[0,0].Left - Shape2.Left)+(i*MatrizEdits[0,0].Width div 2);
    ShapeNiveis[i].Height := 1;
    ShapeNiveis[i].Name := 'ShapeNiveis' + IntToStr(i);
    //cria os labels dos Niveis
    MatrizLabelNiveis[i] := TLabel.Create(Self);
    MatrizLabelNiveis[i].Parent := ScrollBox1;
    MatrizLabelNiveis[i].Width :=
    MatrizLabelNiveis[i].Canvas.TextWidth(IntToStr(i));
    MatrizLabelNiveis[i].Left :=
    ShapeNiveis[i].Left-MatrizLabelNiveis[i].Width;
    MatrizLabelNiveis[i].Top :=
    ShapeNiveis[i].Top - (MatrizLabelNiveis[i].Height div 2);
    MatrizLabelNiveis[i].Caption:=IntToStr(i+1);
   end;
 end;
Shape1.Width:=ShapeEletrodos[3].Left-ShapeEletrodos[0].Left;
Shape2.Height:=MatrizEdits[0,0].Height+5;
end;
```

Fim do Comando 4.9

O `Edit00` foi a primeira componente criada, pois a posição de todos os eletrodos, níveis e até mesmo os outros `edits` do `ScrollBox` são calculados a partir do elemento `MatrizEdits[0,0]`. Portanto, após a construção da primeira caixa de texto, foi escrito o Comando 4.9 para criar os `Shapes` e `labels` dos eletrodos e dos níveis:

O procedimento `TForm2DResIPDadosObs.AcrescentaEletrodo` foi desenvolvido para acrescentar os `edits`, em colunas, nos quais o usuário irá inserir ou ρ_a, ou PFE ou MF, isso dependerá apenas da opção escolhida. Para codificar o procedimento, primeiramente, foram descritas as variáveis `i` e `j` com a função de contadores e `PosX` e `PosY` que têm como obje-

tivo definir a posição das caixas de texto, lembrando que, no Delphi, o sentido crescente do formulário é de cima para baixo para o eixo Y e da esquerda para direita para o eixo X. Em seguida, determinou-se o comprimento dos vetores `ShapeEletrodos`, `MatrizLabelEletrodo` e `MatrizEdits` com o procedimento `SetLength` (Comando 4.10).

```
procedure TForm2DResIPDadosObs.AcrescentaEletrodo;
var
  i,j,PosX,PosY: Integer;
begin
```

Início do Comando 4.10

```
//Define o tamanho do vetor ShapeEletrodos
SetLength(ShapeEletrodos, StrToInt(EditEletrodos.Text));
//Define o tamanho do vetor MatrizLabelEletrodo
SetLength(MatrizLabelEletrodo, StrToInt(EditEletrodos.Text));
//Define o tamanho da matriz MatrizEdits
SetLength(MatrizEdits,StrToInt(EditNiveis.Text),StrToInt(EditEletrodos.Text)
-3);
```

Fim do Comando 4.10

Início do Comando 4.11

```
//Cria as componentes do delphi p/ o formulario
for i:=0 to StrToInt(EditNiveis.Text)-1 do
 begin
  for j:=NodeEditAntes-i to (StrToInt(EditEletrodos.Text)-3)-(1+i) do
   begin
```

Início do Comando 4.11.(a)

```
    if j=0 then
     PosX:=5+MatrizEdits[i-1,j].Left+(MatrizEdits[i-1,j].Width Div 2)
     else
      PosX:=MatrizEdits[i,j-1].Left+MatrizEdits[i,j-1].Width+5;
```

Fim do Comando 4.11.(a)

Início do Comando 4.11.(b)

```
    if i=0 then PosY:=Shape2.Top
     else PosY:=MatrizEdits[i-1,j].Top+MatrizEdits[i-1,j].Height+5;
```

Fim do Comando 4.11.(b)

Início do Comando 4.11.(c)

```
//Cria os Edits
MatrizEdits[i,j] := TEdit.Create(Self);
MatrizEdits[i,j].Parent := ScrollBox1;
MatrizEdits[i,j].Left := PosX;
MatrizEdits[i,j].Top := PosY;
MatrizEdits[i,j].Width := MatrizEdits[0,0].Width;
MatrizEdits[i,j].Height := MatrizEdits[0,0].Height;
MatrizEdits[i,j].Name := 'EditL' + IntToStr(i)+'C'+IntToStr(j);
MatrizEdits[i,j].Text := '';
```

Fim do Comando 4.11.(c)

Início do Comando 4.11.(d)

```
if i=0 then
begin
 //cria os Shapes dos Eletrodos
 ShapeEletrodos[j+3] := TShape.Create(Self);
 ShapeEletrodos[j+3].Parent := ScrollBox1;
 ShapeEletrodos[j+3].Left := Round(ShapeEletrodos[j+2].Left+
 MatrizEdits[0,j].Width+5);
 ShapeEletrodos[j+3].Top := Shape1.top-5;
 ShapeEletrodos[j+3].Width := 1;
 ShapeEletrodos[j+3].Height := MatrizEdits[0,0].top - Shape1.top;
 ShapeEletrodos[j+3].Name := 'ShapeEletrodos' + IntToStr(j+3);
 //cria os labels dos Eletrodos
 MatrizLabelEletrodo[j+3] := TLabel.Create(Self);
 MatrizLabelEletrodo[j+3].Parent := ScrollBox1;
 MatrizLabelEletrodo[j+3].Width :=
 MatrizLabelEletrodo[j+3].Canvas.TextWidth(IntToStr(j+3));
 MatrizLabelEletrodo[j+3].Left := ShapeEletrodos[j+3].Left-
 MatrizLabelEletrodo[j+3].Width;
 MatrizLabelEletrodo[j+3].Top := ShapeEletrodos[j+3].Top -20;
 MatrizLabelEletrodo[j+3].Caption:=IntToStr(j+4);
```

```
    end;
```

Fim do Comando 4.11.(d)

```
   end;
 end;
```

Início do Comando 4.11.(e)

```
Shape1.Width:=ShapeEletrodos[StrToInt(EditEletrodos.Text)-1].Left-
ShapeEletrodos[0].Left;
```

Fim do Comando 4.11.(e)

Início do Comando 4.11.(f)

```
NodeEditAntes:=StrToInt(EditEletrodos.Text)-3;
```

Fim do Comando 4.11.(f)

```
end;
```

Fim do Comando 4.11

Observe no Comando 4.10 que a `MatrizEdits` possui a sua segunda dimensão igual ao número de eletrodos menos três (`StrToInt(EditEletrodos.Text)-3`), devido os `edits` do primeiro nível (Figura 3.14) ficarem entre dois eletrodos, com exceção dos extremos.

As cinco primeiras linhas do Comando 4.11, mostra que após dimensionados os vetores, foram criados os componentes com os laços:

```
for j := NodeE ditAntes-i to (StrToInt(EditEletrodos.Text)-3)-(1+i) do
```

Dentro de:

```
for i:=0 to StrToInt(EditNiveis.  Text)-1 do
```

Onde `i` é o contador para todos os componentes que representam os níveis, ou seja, o número de linhas da matriz, e `j` o número de eletrodos.

Dentro dos laços foi feita a condição abaixo para determinar a posição horizontal dos `edits` (Comando 4.11.(a)).

Logo abaixo do código que determina o local das caixas de texto alinhadas, fez-se o laço para definir o arranjo vertical dos elementos do vetor `MatrizEdits` (Comando 4.11.(b)).

Após condicionado a posição das caixas de texto, foram edificados os `edits` com o Comando 4.11.(c), lembrando que o primeiro já foi criado no `OnCreate` do `Form2DResIPDadosObs`.

Para finalizar o uso do laço de repetição, criaram-se os `Shapes` para representar os eletrodos e os `Labels` para numerá-los quando `i = 0`, baseando as posições das componentes com as dos `edits`. veja o Comando 4.11.(d).

Para representar a distribuição dos eletrodos na superfície da terra foi utilizada a componente com a propriedade `name` igual a `Shape1`. A única alteração que ela sofre em tempo de execução é o seu comprimento, como mostra o Comando 4.11.(e), pois deve ser sempre igual à diferença entre o `left`(abscissa da extremidade esquerda da componente no formulário) do último eletrodo e o primeiro.

E para concluir o procedimento, é necessário registrar a quantidade de `EditEletrodos` que há na tela, para quando o `TForm2DResIPDadosObs.AcrescentaEletrodo` for acionado novamente, atualize a janela do IP - Resistividade com novas caixas de edição. Portanto, fez-se Comando 4.11.(f).

Depois dos comandos para adicionar `Shapes`, `Labels` e `Edits` na horizontal, foi necessário desenvolver um procedimento para excluir as componentes. Assim, fez-se `TForm2DResIP DadosObs.ExcluiEletrodo`, cuja sua codificação é o Comando 4.12.

Início do Comando 4.12

```
procedure TForm2DResIPDadosObs.ExcluiEletrodo;
var
  i,j: Integer;
begin if StrToInt(EditEletrodos.Text)>=4 then
 begin
  for i:=0 to StrToInt(EditNiveis.Text)-1 do
   begin
    for j:=NodeEditAntes-(1+i) downto StrToInt(EditEletrodos.Text)-(i+3) do
     begin
      MatrizEdits[i,j].Free;
       if i=0 then
        begin
         ShapeEletrodos[j+3].Free;
         MatrizLabelEletrodo[j+3].Free;
```

```
        end;
      end;
    end;
  Shape1.Width:=
  ShapeEletrodos[StrToInt(EditEletrodos.Text)-1].Left-ShapeEletrodos[0].Left;
  NodeEditAntes:=StrToInt(EditEletrodos.Text)-3;
 end;
end;
```

Fim do Comando 4.12

O objetivo de fazer as ações para acrescentar e diminuir, horizontalmente, os `Shapes`, `Labels` e `Edits`, foram para serem chamadas no evento `OnChange` do `EditEletrodos`, mas antes foi necessário fazer um laço condicional para o programa satisfazer a equação 4.1. Portanto, fez-se o Comando 4.13.

```
procedure TForm2DResIPDadosObs.EditEletrodosChange(Sender:TObject);
Var
  No,Cod,QuantMaxNi,QuantMinEl:Integer;
begin if EditEletrodos.Text<>'' then
 begin
  Val(EditEletrodos.Text,No,Cod);
  if (Cod <>0) or (No=0) then
   begin
    MessageDlg('O Numero de Camadas deve ser entre 1 e 100',mtInformation,
    [mbOK],0);
    EditEletrodos.Text:='';
    EditEletrodos.SetFocus;
   end
  else if No<=100  then
   begin
    NEletrodos:=StrToInt(EditEletrodos.Text);
    Niveis:=StrToInt(EditNiveis.Text);
```

Início do Comando 4.13

```
    QuantMaxNi:= Trunc((StrToInt(EditEletrodos.Text)-1) div 2);
```

```
    if EditNiveis.Text='1' then QuantMinEl:=4
    else QuantMinEl:= 2*StrToInt(EditNiveis.Text)+1;
```

Fim do Comando 4.13

Início do Comando 4.14

```
    if StrToInt(EditEletrodos.Text) >= QuantMinEl then
     begin
      if (StrToInt(EditEletrodos.Text)-3)> NodeEditAntes then
       AcrescentaEletrodo
      else if (StrToInt(EditEletrodos.Text)-3)< NodeEditAntes then
       ExcluiEletrodo;
     end
    else
     begin
      if (StrToInt(EditEletrodos.Text)-3)> NodeEditAntes then
       AcrescentaEletrodo
       else if (StrToInt(EditEletrodos.Text)-3)< NodeEditAntes then
        ExcluiEletrodo;
      if StrToInt(EditEletrodos.Text)>=4 then
       EditNiveis.Text:=IntToStr(QuantMaxNi);
     end;
```

Fim do Comando 4.14

```
    ScrollBox1.ClientWidth:=Shape1.Width+Shape1.Left+10;
    StsBar1.Panels[0].Text:='N° de Eletrodos: '+ EditEletrodos.Text+';'+
                            ' N° de Niveis: '+EditNiveis.Text;
   end
       else
        begin
         MessageDlg('O Numero de Camadas deve ser entre 1 e 100',
         mtInformation,[mbOK],0);
         EditEletrodos.SetFocus;
        end;
 end;
end;
```

Onde, no procedimento `TForm2DResIPDadosObs.EditEletrodosChange`, `QuantMaxNi` é a quantidade máxima de níveis, e `QuantMinEl` é a quantidade mínima de eletrodos. Sabendo os limites de eletrodos e níveis, foi possível fazer o laço para acrescentar linhas caso a quantidade mínima de eletrodos extrapole, aumentando desse modo a quantidade máxima de eletrodos. Veja Comando 4.14.

Para desenvolver o evento `OnChange` da componente `EditNiveis`, bastou estruturar usando o mesmo raciocínio para `EditEletrodos`, ou seja, fez-se procedimentos para acrescentar e diminuir `Edits`, `Shapes` e `Labels` em colunas, chamados de, respectivamente, `AcrescentaNiveis` e `ExcluiNiveis`, depois foram definidas duas variáveis para determinar o limite máximo de níveis e mínimo de eletrodos e, por fim, o laço condicional do Comando 4.15.

```
procedure TForm2DResIPDadosObs.EditNiveisChange(Sender: TObject);
Var
  No,Cod,QuantMaxNi,QuantMinEl:Integer;
begin if EditNiveis.Text<>'' then
 begin
  Val(EditNiveis.Text,No,Cod);
  if (Cod <>0) or (No=0) then
   begin
    MessageDlg('O Numero de Camadas deve ser entre 1 e 100',mtInformation,
    [mbOK],0);
    EditNiveis.Text:='';
    EditNiveis.SetFocus;
   end
  else if No<=100  then
   begin
    NEletrodos:=StrToInt(EditEletrodos.Text);
    Niveis:=StrToInt(EditNiveis.Text);
    QuantMaxNi:= Trunc((StrToInt(EditEletrodos.Text)-1) div 2);
    if EditNiveis.Text='1' then QuantMinEl:=4
    else QuantMinEl:= 2*StrToInt(EditNiveis.Text)+1;
```

Início do Comando 4.15

```
    if (StrToInt(EditNiveis.Text) <= QuantMaxNi) and
```

```
      (StrToInt(EditNiveis.Text)<> 1) then
    begin
     if StrToInt(EditNiveis.Text)>NodeNiveisAntes then AcrescentaNiveis
     else if StrToInt(EditNiveis.Text)<NodeNiveisAntes then ExcluiNiveis;
    end
   else
    begin
     if StrToInt(EditNiveis.Text)>NodeNiveisAntes then AcrescentaNiveis
     else if StrToInt(EditNiveis.Text)<NodeNiveisAntes then ExcluiNiveis;
     EditEletrodos.Text:=IntToStr(QuantMinEl);
    end;
```

Fim do Comando 4.15

```
   StsBar1.Panels[0].Text:='N° de Eletrodos: '+ EditEletrodos.Text+';'+
                          ' N° de Niveis: '+EditNiveis.Text;
  end
      else
       begin
        MessageDlg('O Numero de Camadas deve ser entre 1 e 100',
        mtInformation,[mbOK],0);
        EditEletrodos.SetFocus;
       end;
 end;
end;
```

Um detalhe importante tanto para os formulários *dados observados* quanto para os *modelos geoelétricos* é o armazenamento de pontos flutuantes, pois o Delphi para ler um arquivo de dados que serão atribuídos para uma variável real aceita somente ponto (·) como separador decimal, mas no momento de salvar números decimais de uma componente textual para um arquivo de texto, é somente aceito o decimal definido no sistema operacional pelo usuário em *opções regionais* do *painel de controle* do *Windows*, que no Brasil é vírgula(,). O problema foi resolvido fazendo uma função em que troca vírgula por ponto, como pode ser visto no Comando 4.16.

Início do Comando 4.16

```
function TForm00_MenuPrincipal.MudaDecimal(Valor : String):String;
var
   i:integer;
begin i := POS(',',Valor);//Obtem a posicao da virgula While i
> 0 do//repete a operacao enquanto tiver virgula na
variavel
 begin//<=-----------------------------------------------------------|
  Delete(Valor, I ,1);//Exclui a virgula da variavel                  |
  Insert('.',Valor,I);//Inclui um ponto no lugar que a virgula foi tirada|
  I := POS(',',Valor);//obtem a posicao da proxima virgula, se existir   |
  Delete(Valor, I ,1);//Exclui a proxima virgula da variavel          |
 end;//<=-----------------------------------------------------------|
Result := Valor;//Devolve o novo valor com ponto no lugar da virgula.
end;
```

Fim do Comando 4.16

Desse modo, caso haja vírgula(,) no valor do arquivo de leitura, a função converte para ponto(.) e ao salvar é retirada a vírgula e inserido ponto nos dados armazenados.

4.2 - O USO DAS DLLS

No *Windows* existem dois tipos de arquivos executáveis: aplicações e dlls (Dynamic Link Libraries). Normalmente, as aplicações possuem no seu nome a extensão *.EXE e as dlls são caracterizadas por *.DLL, mas é possível usar outras extensões para as bibliotecas, sem haver danos na sua utilização.

A diferença entre um programa executável e as dlls é que o primeiro é ativo, possuindo uma dinâmica própria, ou seja, existe um processo em execução o qual tem início, meio e fim. Já as dlls não, pois são módulos passivos. Tipicamente, uma dll possui uma série de funções e procedimentos que podem ser compartilhados por outros módulos, mas esses códigos ficam desativados e só adquirem uma finalidade quando são chamados por algum programa ou por outra dll. Múltiplos programas podem usar uma mesma biblioteca.

A modelagem numérica (fase de processamento matemático) feito na eGs é todo em linguagem FORTRAN, porém os projetos são compilados no formato de dll, facilitando a implementação de diferentes métodos geofísicos. O primeiro processo que o programa, em

fase de execução, faz para a modelagem numérica é ativar a função `ChamaDLLFortran`, que possui um parâmetro definido por `CaminhodaDLL`, o qual serve para explicitar o diretório da dll, e outro parâmetro descrito pela variável `NomedaFuncao`, para carregar a função contida na biblioteca que se deseja usar. Todos os dois são do tipo `PChar`. Veja o procedimento completa no Comando 4.17.

Início do Comando 4.17

```
Procedure TForm00_MenuPrincipal.ChamaDLLFortran(CaminhodaDLL,
NomedaFuncao:PChar);
type
    TFuncaodaDLL = procedure();//definicao da funcao na DLL
var
   Endereco: THandle;//endereco na memoria(No inteiro)
   Ponteiro: TFarProc;//ponteiro para o endereco da memoria
   MinhaFuncao: TFuncaodaDLL;//variavel do tipo da funcao da DLL
begin
Endereco := LoadLibrary (CaminhodaDLL);//Le a DLL no Caminho especificado
if Endereco <> 0 then//<=-----------------------------------------------|
 try//<=-------------------------------------------------------------||
  Ponteiro := GetProcAddress (Endereco,PChar (NomedaFuncao));         //||
  if Ponteiro <> nil then//<=-----------------------------------------|||
   begin                                                            //|||
    MinhaFuncao := TFuncaodaDLL (Ponteiro);                         //|||
    MinhaFuncao;                                                    //|||
   end                                                              //|||
  else MessageDlg('NAO FOI ACHADA A FUNCAO "'+NomedaFuncao+'" NA DLL'//|||
       ,mtError,[mbOK],0);//<=---------------------------------------|||
 finally                                                             //||
  FreeLibrary (Endereco);                                            //||
 end//<=-------------------------------------------------------------||
else MessageDlg('NAO FOI ACHADA A DLL EM: "'+CaminhodaDLL+'"'          //|
     ,mtError,[mbOK],0);//<=-------------------------------------------|
end;
```

Fim do Comando 4.17

No escopo da função foi declarado um tipo `procedure` chamado de `TFuncaodaDLL` para reconhecer a função desejada na dll com a variável `MinhaFuncao`. `Endereco` serve para armazenar o `Handle` da biblioteca na memória, podendo assim com `Ponteiro` localizá-la.

Portanto, a dll é lida com a função `LoadLibrary (CaminhodaDLL)` e guardada em `Endereco`. Caso `Endereco` seja diferente de zero, isso indica que foi atribuído a ela um local na memória, senão é mostrada uma janela de mensagem informando que não foi possível achar a dll. Uma vez carregada a dll, a função pode ser obtida com `GetProcAddress (Endereco, PChar (NomedaFuncao))`, a qual é possível achá-la no destinatário `Ponteiro`. Se o destino existir, `Ponteiro` não será nulo (`nil`), podendo, portanto, rodar a função desejada na dll, caso contrário, será exibida uma mensagem avisando a ausência da função na biblioteca. Para liberar a memória, usou-se `FreeLibrary (Endereco)`.

Para utilizar a `ChamaDLLFortran` é preciso saber como é a metodologia da dll. A função inserida na biblioteca lê um arquivo com extensão *.$$$, nele está o caminho dos arquivos *modelos geoelétricos*, *dados observados* e os diretórios de saída da modelagem numérica. Para isso, fez-se uma ação no `ActionManager1` com a propriedade `name` igual a `Executar`, cuja ativação ocorre somente quando existe uma janela "Modelo" e outra "Dados observados" abertas.

A ação `Executar` tem a finalidade de verificar se existem os arquivos do *modelo geoelétrico* e *dados observados*, atualizar os diretórios em *.$$$ e ativar a dll. Para melhor esclarecimento do procedimento, será comentado, passo a passo, a implementação do método VLF (Very Low Frequency) em 2D. Veja:

1° Passo) Verificar *.MGE e *.OBS: Nesta etapa o programa averigua se as variáveis `ModqueVaiproDolar` e `ObsqueVaiproDolar1` estão vazias, se for verdade são criados arquivos com o nome "Temporário" no mesmo local da eGs, referentes à janela "Modelo" e "Dados observados". Caso contrário, o diretório usado é o mesmo salvo pelo o usuário. Veja um trecho da implementação da modelagem VLF em 2D no Comando 4.18.

Início do Comando 4.18

```
procedure TForm00_MenuPrincipal.ExecutarExecute(Sender: TObject);
const
  TEMPMOD='Temporario.MOE';
  TEMPOBS='Temporario.OBS';
var
```

```
  ArquivoDolar1,ArquivoDolar2:TextFile;
begin

                                        ⋮

if Form00_Arvore.RadioButton2D.Checked then
 begin
  Case Form00_Arvore.ItemdaArvore of
  //-----Very Low frequency (VLF)----------------\\
   4:
    begin
     try
      if (ModqueVaiproDolar='') then
       begin
        ModqueVaiproDolar:=ExtractFilePath(ParamStr(0)) +TEMPMOD;
        Form4_Malha3D.Salvar(ModqueVaiproDolar);
       end;
      if (ObsqueVaiproDolar1='') then
       begin
        ObsqueVaiproDolar1:=ExtractFilePath(ParamStr(0))+TEMPOBS;
        Form2DVLFDadosObs.Salvar(ObsqueVaiproDolar1);
       end;
```

Início do Comando 4.18.(a)

```
      AssignFile(ArquivoDolar1,'c:\egs\2D\VLF\PD\vlf2d.$$$');
      Rewrite(ArquivoDolar1);
      writeln(ArquivoDolar1, ObsqueVaiproDolar1);
      writeln(ArquivoDolar1, ModqueVaiproDolar);
      Writeln(ArquivoDolar1,CriaSubDir('ELP'));
      Writeln(ArquivoDolar1,CriaSubDir('Hx0'));
      Writeln(ArquivoDolar1,CriaSubDir('Hz0'));
```

Fim do Comando 4.18.(a)

Início do Comando 4.18.(b)

```
    finally
     CloseFile(ArquivoDolar1);
     ////////
     ChamaDLLFortran('c:\egs\2D\VLF\PD\vlf2d_dll.dll','VLF2D');
     ///////

                                    ⋮

     end;
```

Fim do Comando 4.18.(b)

```
end;

                                    ⋮

end;
```

Fim do Comando 4.18

`ModqueVaiproDolar` e `ObsqueVaiproDolar1` são `string` globais que têm por finalidade atribuir o caminho do arquivo para os *modelos geoelétricos* e *dados observados*. Para acrescentar as arquivos no diretório da eGs foi usada a função `ExtractFilePath`, cuja resposta é o caminho da eGs, e a constante `TEMPMOD` para nomear os arquivos temporários.

2º Passo) Atualizar os diretórios em *.$$$: Os arquivos com a extensão *.$$$ têm a finalidade de informar à dll onde está o *modelo geoelétrico*, os *dados observados* e qual o diretório que se deseja salvar os arquivos de saída.

O intuito inicial era criar o arquivo *.$$$ em qualquer lugar do disco rígido, porém no mesmo diretório da dll, mas isso não foi possível devido a um conflito entre a propriedade `FileName` da componente `OpenDialog` do Delphi com o comando `open` do FORTRAN, ambos usados para chamar arquivos. Portando, teve-se que fixar os caminhos do *.$$$ em subdiretórios da eGs, seguindo os seguintes critérios:

- A eGs deve ser instalada na unidade C:\ dentro de uma pasta chamada eGs, ou seja, para encontrar o aplicativo o caminho é C:\eGs\eGs.EXE;
- Os subdiretórios devem seguir a ordem da árvore da janela Métodos, por exemplo, o arquivo *.$$$ e a dll para o uso da VLF no problema direto em duas dimensões necessitam estar em C:\eGs\2D\VLF\PD.

Voltando para a construção da ação `Executar`, usando a implementação da dll VLF2D. Para atualizar *.$$$, desenvolveu-se o Comando 4.18.(a).

Onde `AssignFile` cria o arquivo vlf2d.$$$, `Rewrite` abre para editar e `writeln` escreve, nessa ordem, o caminho do arquivo para os *dados observados*, *modelos geoelétricos* e os de saída, que no caso do VLF são a elipsidade (*.ELP), componente do campo magnético horizontal (*.Hx0) e componente do campo magnético vertical (*.Hz0).

Decidiu-se criar os arquivos de saída no mesmo caminho do *modelos geoelétricos*, porém em uma subpasta. Logo, fez-se o procedimento `CriaSubDir` que tem como parâmetro a extensão do arquivo, como é possível ver nas três últimas linhas do comando Comando 4.18.(a).

3º Passo) Ativar a dll: Para utilizar a dll basta usar a função `ChamaDLLFortran`.

No exemplo ilustrativo é o Comando 4.18.(b).

Assim, a estruturação para o uso da dll está completo e, conseqüentemente, a fase de pré-processamento e sua ligação com o processamento foi finalizada.

5 - CONCLUSÕES E SUGESTÕES

Os métodos elétricos e eletromagnéticos são amplamente usados em geofísica, cujo principal instrumento, após o levantamento de campo, é o computador, especialmente nos dias atuais em que as máquinas estão ficando cada vez mais rápidas e com custo mais baixo, permitindo, desse modo, o processamento e a interpretação de dados de levantamento de campo mais precisa e mais rápida. Assim, com a finalidade de agilizar o trabalho da geofísica de exploração por meio dos métodos elétricos e eletromagnético, desenvolvolveu-se para *Windows* a eGs (*electrical Geophysics suite*), uma ferramenta computacional "amigável" e de fácil uso. A eGs é um software que possibilita inserir dados no computador, fazer a modelagem numérica, inversão automática e a visualização para interpretação em um tempo consideravelmente pequeno, se comparado ao mesmo serviço feito antes da sua criação. A eGs se destina, principalmente, ao meio acadêmico como um complemento ao ensino das disciplinas dos métodos elétricos e eletromagnéticos nos cursos de graduação e pós-graduação em geologia e geofísica. Serve também para auxiliar os trabalhos de conclusão de curso (TCC) e dissertações de mestrado. Obviamente, ela pode também ser de grande utilidade na indústria.

Com o objetivo de verificar se a interface gráfica eGs facilita o pré-processamento, processamento e pós-processamento, foram feitos vários testes com dados sintéticos no capítulo 3. Outra verificação da utilidade do aplicativo é o trabalho apresentado por Gilberto E. Reis Vogado em sua dissertação de mestrado (Vogado, 2005), sob a orientação do Prof. Om Prakash Verma, que consiste em usar a eGs em uma aplicação do método VLF (Very Low Frequency) em duas dimensões. Além disso, muitas teses já foram realizadas usando as versões anteriores da eGs (veja as referências bibliograficas). Com isso, mostra-se que o programa é satisfatório ao meio acadêmico.

Observa-se, com o histórico da ciência Geofísica, que ela está sempre procurando melhorar os métodos existentes e construir novas técnicas para exploração, e a tecnologia computacional está constantemente em desenvolvimento, portanto, é fácil constatar que um aplicativo como a eGs deve ser permanentemente atualizado. No Capítulo 4, foram explicados muitos detalhes para aqueles interessados no desenvolvimento da eGs.

Existem ainda muitos aprimoramentos, aos quais não foram feitos, pois são detalhes que não impedem o funcionamento do programa. O primeiro aperfeiçoamento que deve ser feito é instalar o aplicativo em qualquer diretório, pois não foi possível compreender o conflito entre o FORTRAN e o Delphi em tempo hábil. Outra adição no desenvolvimento é a construção

de um código para salvar os arquivos do *modelo geoelétrico* e dos *dados observados*, de modo que ao serem abertos, inicie a eGs. E por fim, criar uma versão para o sistema operacional Linux, pois os software livres estão começando ser bastante utilizados e, afinal de contas, a eGs é um software livre.

REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, F. L., 2002. **Filtros otimizados para as transformadas seno, cosseno e de Hankel J_0, J_1 e J_2**. Belém, Universidade Federal do Pará Centro de Geociências. 93 p. (Dissertação de Mestrado)

BECKER, E. B., CAREY, G. F. E, ODEN, J. T., 1981. **Finite Elements - An introduction**. New Jersey, Prentice-Hall. 258 p.

CAGNIARD, L., 1953. Basic theory of the magnetotelluric method of geophysical prospecting. **Geophysics**, 18: 605-635.

CAMPBELL, W. H., 1989. An introduction to quiet daily geomagnetics fields. **Pure and Applied Geophysics**, 131: 315-331.

CANTÙ, M., 2002. **Mastering delphi 6**. San francisco, Sybex. 1071 p.

CARRASQUILLA, A. A. G., 1984. **Estudo geofísico regional sobre as águas subterrâneas na Ilha de Marajó Pará - Brasil**. Belém, Universidade Federal do Pará. Centro de Geociências. 143 p. (Dissertação de Mestrado)

CARRASQUILLA, A. A. G., 1993. **Modelagem numérica da influência do eletrojato equatorial em dados magnetotelúricos produzidos por estruturas tridimensionais**. Belém, Universidade Federal do Pará. Centro de Geociências. 176 p. (Tese de Doutorado)

GUIMARÃES, P. B. C., 1983. **Efeitos do strike das estruturas geológicas de duas dimensões nas pseudo-seções de IP e resistividade**. Belém, Universidade Federal do Pará. Centro de Geociências. 148 p. (Dissertação de Mestrado)

LUIZ, J. G., 1999. **Informação a priori na inversão de dados magnetotelúricos bidimensionais**. Belém, Universidade Federal do Pará. Centro de Geociências. 93 p. (Tese de Doutorado)

LUIZ, J. G. & SILVA, L. M. C., 1995. **Geofísica de prospecção. Belém**. Editora Universitária, 311 p.

MONTEIRO, H. D. M., 2001. **Interface gráfica para pré e pós-processamentos de dados elétricos e eletromagnéticos em geofísica**. Belém, Universidade Federal do Pará. Centro de Geociências. 56 p. (Dissertação de Mestrado)

MOTA, J. P., 1991. **Avaliação da influência do eletrojato equatorial na interpretação de dados magnetotelúricos bidimencionais**. Belém, Universidade Federal do Pará. Centro de Geociências. 93 p. (Dissertação de Mestrado)

RIJO, L., 1992. Notas de aula do Curso de Pós-Graduação em Geofísica, UFPA. Belém (www.rijo.pro.br).

SCHMITZ, E. A. & TELES, A. A. S., 1989. **Pascal e técnicas de programação**. Rio de janeiro, LTC - Livros Técnicos e Científicos Editora S.A.. 287 p.

SILVA, M. W., 2005. **Influencia de estruturas geológicas bidimensionais no campo geoeletromagnético na presença do eletrojato equatorial**. Belém, Universidade Federal do Pará. Centro de Geociências. 121 p. (Tese de Doutorado)

SILVA, R. E. C., 2005. **Interfaces gráficas de entrada e saída de dados da eGs (electral Geophysics suite)**. Belém, Universidade Federal do Pará. Centro de Geociências. 186 p. (Dissertação de Mestrado)

TIKHONOV, A. N., 1950. On determining electrical characteristics of the deep layers of the earth's crust. **Doklady**, 73: 295-297.

VOGADO, G. E. R., 2005. **Estudo de efeito de manto no VLF através de modelagem numérica**. Belém, Universidade Federal do Pará. Centro de Geociências. Dissertação de Mestrado em Geofísica (em fase de conclusão).

APÊNDICE A – MANUAL DO USUÁRIO

A melhor maneira de se familiarizar com um novo programa é através de exemplos. Por isso serão dados dois exemplos explorando algumas opções da eGs. Será iniciado com um exemplo de modelagem 1D e depois de modelagem 2D.

A.1 – ONDE OBTER A eGs 3.0

A eGs 3.0 e a 1ª versão do manual serão distribuídos através do site de professor Luiz Rijo cujo endereço é *www.rijo.pro.br*.

Este manual, assim como a eGs, serão periodicamente atualizados com sugestões dos usuários e com implementação de novas opções de métodos elétricos e eletromagnéticos.

É importante informar que a eGs é um software livre, ou seja sem fins lucrativos, portanto é de suma importância a colaboração dos usuários para melhorar o desempenho do aplicativo.

A.2 – INSTALAÇÃO DA eGs 3.0 (Windows 2000 ou XP)

Para iniciar o processo de instalação do aplicativo, o usuário deve seguir os seguintes passos:

1º) Verifique se existe alguma versão anterior da eGs;

2º) Verifique se existe espaço suficiente em disco. A instalação completa da eGs ocupa entre 4 MB a 6MB de espaço no disco, isto vai depender da versão do Windows;

3º) Escolha o local de pasta para instalação c:\eGs. Este diretório irá receber todos os arquivos da distribuição;

4º) Inicie eGs setup wizard (setup_eGs.exe, mostrado na Figura A.1). Será apresentada a janela de boas-vindas (Figura A.2);

Figura A.1: Ícone de instalação da eGs 3.0.

eGs 3.0 Install Program
Welcome to the eGs 3.0 Install program.
This program allows you to install eGs 3.0 on your hard drive.
It is strongly recommended that before proceeding, you ensure that no other Windows programs are running.
If you do not wish to install eGs 3.0, click 'Exit' now, otherwise click 'Next' to continue.
< Back
Next >
Exit

Figura A.2: Janela de boas-vindas para instalação do programa

5º) Clique em "*Next*" para ir a próxima janela. A Figura A.3 mostra informação do local onde a eGs deve ser instalada;

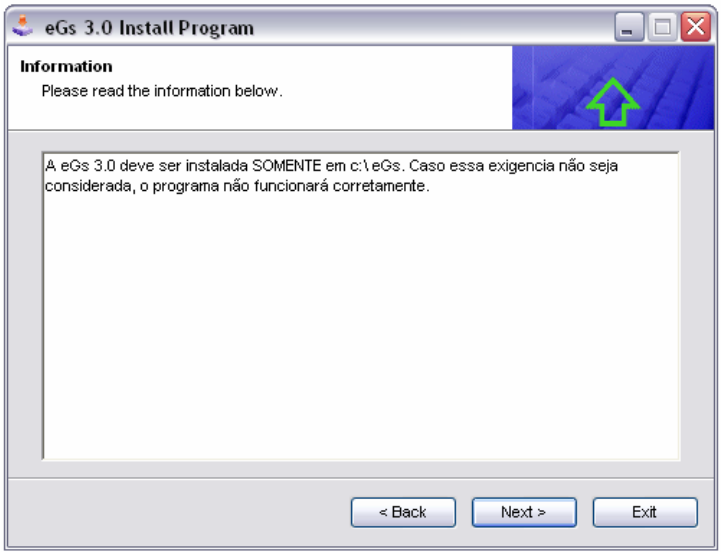

eGs 3.0 Install Program
Information
Please read the information below.
A eGs 3.0 deve ser instalada SOMENTE em c:\eGs. Caso essa exigencia não seja considerada, o programa não funcionará corretamente.
< Back
Next >
Exit

Figura A.3: Informação para instalação.

6º) Clique em "*Next*" para ir a próxima janela. A Figura A.4 mostra informações sobre o termo de licença para ser lido pelo usuário. Após lido, deve-se selecionar "*I agree with the above terms and conditions*" para concordar e "*I do not agree*" para não concordar;

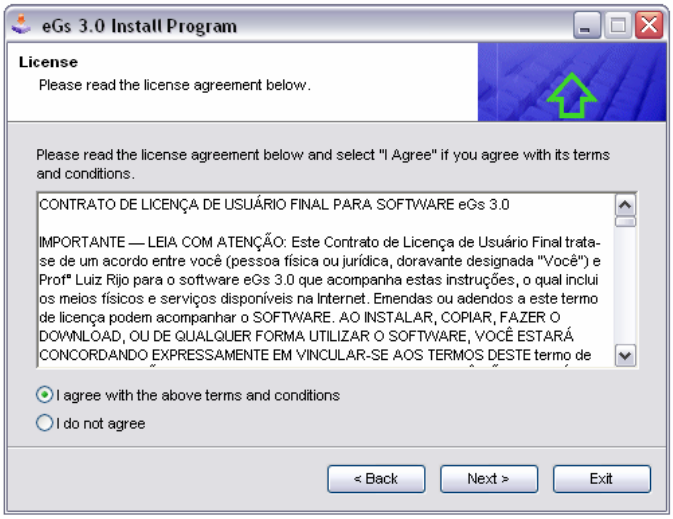

eGs 3.0 Install Program
License
Please read the license agreement below.
Please read the license agreement below and select "I Agree" if you agree with its terms and conditions.
CONTRATO DE LICENÇA DE USUÁRIO FINAL PARA SOFTWARE eGs 3.0
IMPORTANTE — LEIA COM ATENÇÃO: Este Contrato de Licença de Usuário Final trata-se de um acordo entre você (pessoa física ou jurídica, doravante designada "Você") e Prof° Luiz Rijo para o software eGs 3.0 que acompanha estas instruções, o qual inclui os meios físicos e serviços disponíveis na Internet. Emendas ou adendos a este termo de licença podem acompanhar o SOFTWARE. AO INSTALAR, COPIAR, FAZER O DOWNLOAD, OU DE QUALQUER FORMA UTILIZAR O SOFTWARE, VOCÊ ESTARÁ CONCORDANDO EXPRESSAMENTE EM VINCULAR-SE AOS TERMOS DESTE termo de
I agree with the above terms and conditions
I do not agree
< Back
Next >
Exit

Figura A.4: Termo de licença.

7º) Se o termo de licença estiver sido concordado, clique em "*Next*" para ir a próxima janela. A Figura A.5 mostra a informação sobre o local onde a eGs será instalada, que dever ser em c:\eGs, pois em outra pasta o aplicativo não funcionará. Abaixo do diretório de instalação, estão as informações sobre o espaço necessário e o espaço disponível em disco;

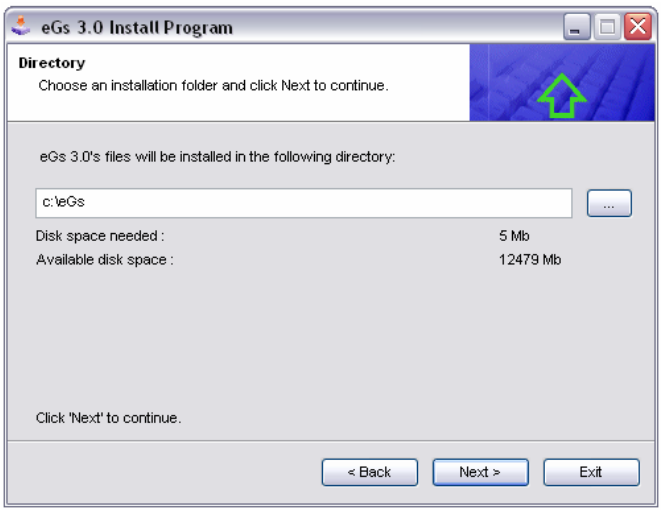


Figura A.5: Informações do diretório, do espaço necessário e livre no disco rígido.

8º) Clique em "*Start*" para instalar o programa. A Figura A.6 mostra a janela de confirmação de instalação no diretório c:\eGs.

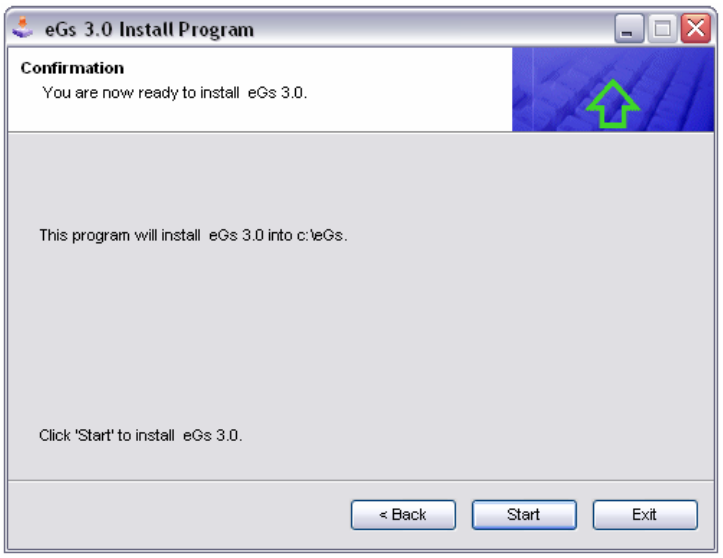


Figura A.6: Janela de confirmação da instalação.

9º) Clique em "*Next*" para finalizar a instalação. A Figura A.7 mostra a janela informativa de sucesso da instalação do programa.

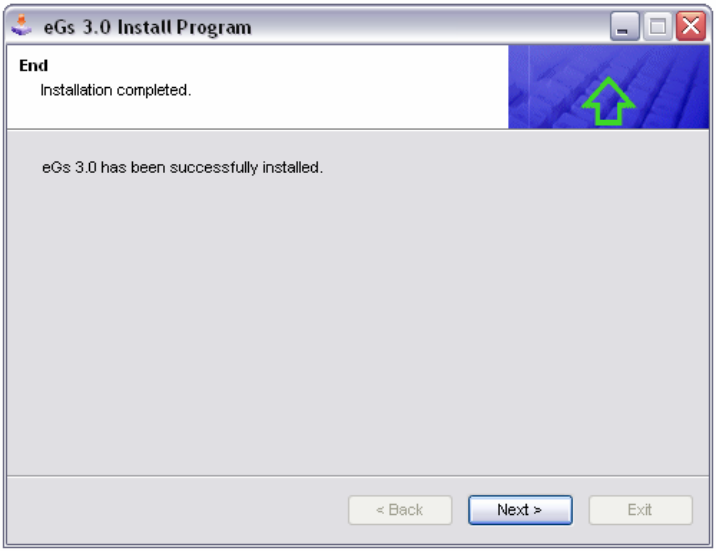


Figura A.7: Janela que informa a finalização da instalação do programa.

A.3 – COMO USAR A eGs 3.0

A.3.1 – 1D (uma dimensão)

A seguir, vamos mostrar um exemplo de modelagem 1D usando o método magnetotelúrico.

A.3.1.1 – Magnetotelúrico (Problema Direto)

Para utilizar a eGs em uma aplicação geofísica que utiliza o método magnetotelúrico, faz-se os seguintes passos:

1º) Escolher o tipo de problema.

Para construir um novo ou abrir um arquivo ao modelo, primeiramente é necessário escolher, na janela "Métodos" (Figura A.8), a dimensão, em seguida dê dois clique em ⊞ Sondagem Magnetotelúrica e o tipo de problema (direto ou inverso) que deseja usar.

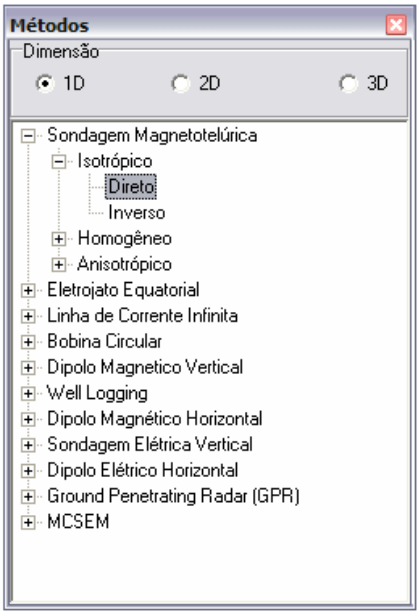


Figura A.8: Janela Métodos

2º) Construir um *modelo geoelétrico*.

Depois de feita a escolha do tipo de problema, há duas maneiras de editar um modelo geoelétrico:

i. A partir da edição de um *modelo geoelétrico* novo;
ii. A partir de um *modelo geoelétrico* salvo anteriormente.

i) A partir da edição de um modelo geoelétrico novo:

Para abrir uma janela "Modelo" para editar um modelo estratificado em camadas, primeiramente deve ser escolhido o tipo de problema na janela "Métodos", depois basta seguir um dos dois caminhos seguintes:

- Janela Seleção → Modelo → Novo (Figura A.9);
- Menu → Arquivo → Novo → Modelo (Figura A.10).

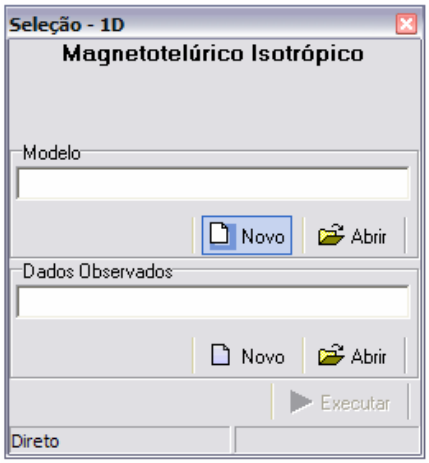


Figura A.9: Janela "Seleção" para abrir um *modelo geoelétrico* novo.

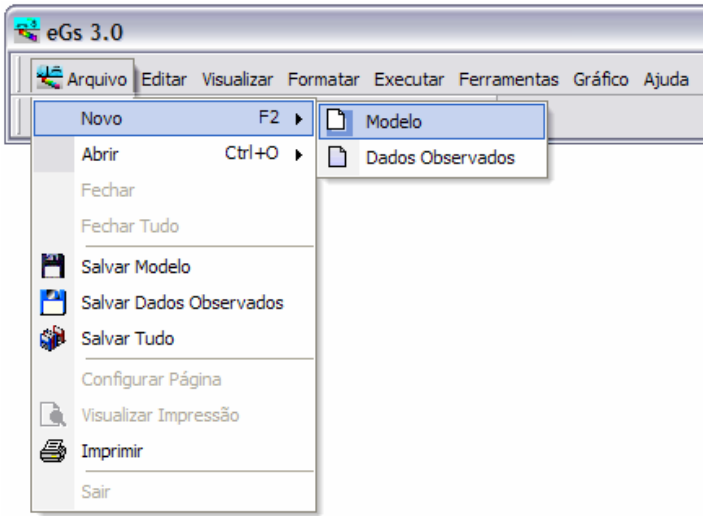


Figura A.10: Abrir um *modelo geoelétrico* novo através do Menu.

Surgirá a janela "Modelo" mostrada na Figura A.11.

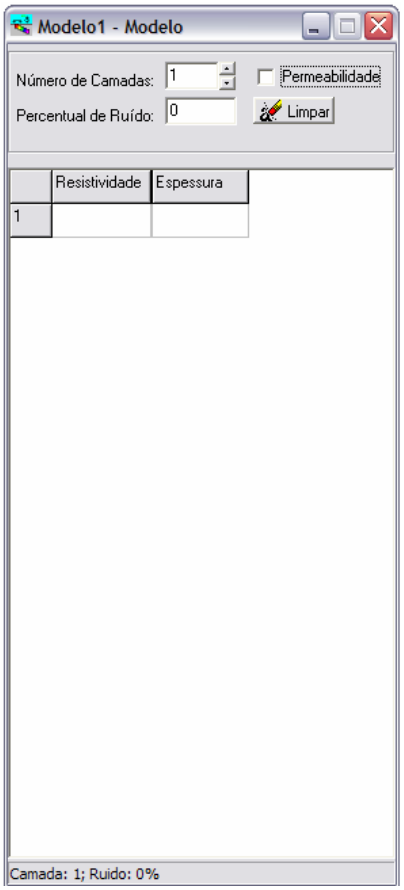


Figura A.11: Janela "Modelo" para início da edição de um *modelo geoelétrico*.

Como exemplo, usou-se um meio estratificado com cinco camadas, como mostra a Figura A.12.

Resistividade ↓	Permeabilidade magnética relativa ↓	Espessura ↓
100 ohm-m	1	1 km
200 ohm-m	1	2 km
2 ohm-m	1	1 km
1000 ohm-m	1	30 km
10 ohm-m	1	

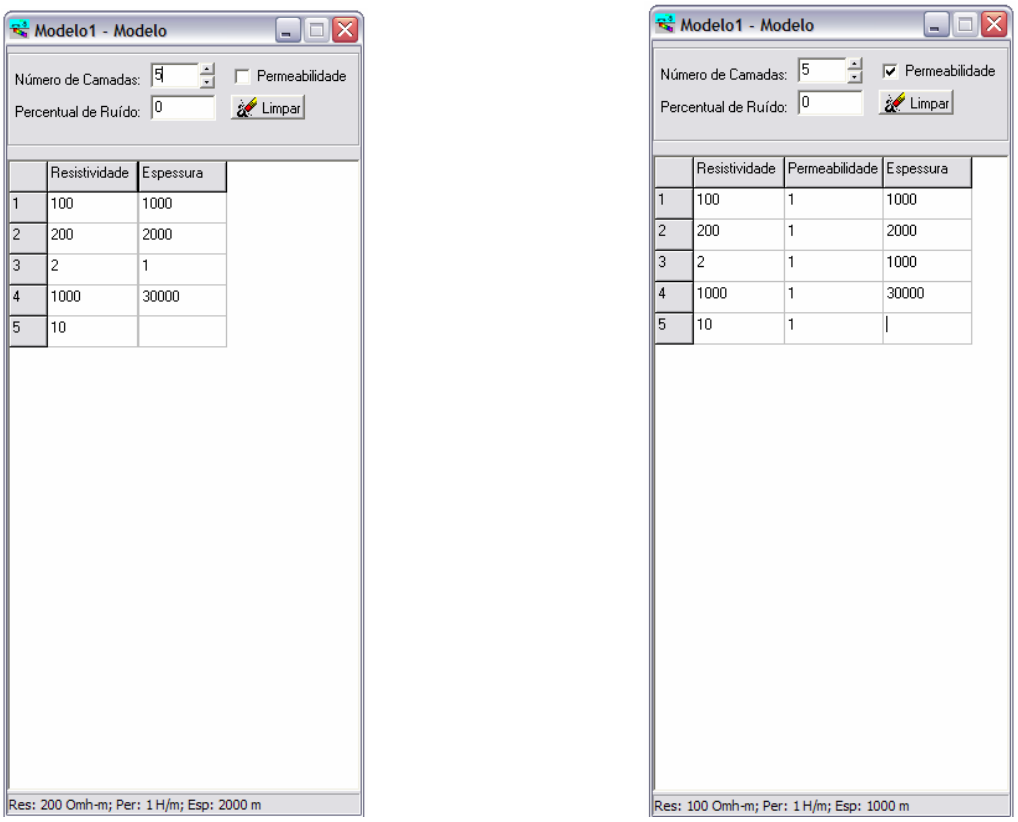
Figura A.12: *Modelo geoelétrico* 1D usado como exemplo.

Observe na Figura A.12 que a ultima camada é o substrato, não sendo necessário ser editada, portanto a eGs não permite editar a espessura do substrato.

As Figuras A.13.(a) e A.13.(b) são as janelas "Modelo", as quais servem para editar o *modelo geoelétrico*, onde:

⇒ O *Número de camadas* altera o número de linhas da tabela da janela "Modelo", podendo ser através do teclado digitando valores entre 1 e 100 inclusive ou pelo mouse através de clique nas setas. Se o clique for na seta apontada para cima, os valores são aumentados de um em um, e se para baixo os valores são diminuídos;

⇒ A primeira e segunda colunas da grade, na Figuras A.13.(a), representam, respectivamente, resistividade (ohm-m) e espessura (m) das camadas.



(a) (b)

Figura A.13: (a) Janela "Modelo" com permeabilidade relativa igual a 1 em todas camadas. (b) Janela "Modelo" para alteração da permeabilidade relativa.

A eGs usa como padrão a permeabilidade relativa igual a 1. Caso o usuário deseje alterar a permeabilidade relativa, basta marcar o item *permeabilidade*, e uma coluna chamada permeabilidade será mostrada entre as colunas resistividade e espessura, como mostra a Figura A.13.(b).

Para salvar o *modelo geoelétrico* pode ser de duas maneiras: através do item salvar modelo do menu (Figura A.14) ou quando o usuário desejar fechar a janela "Modelo" e não salvou as alterações (Figura A.15);

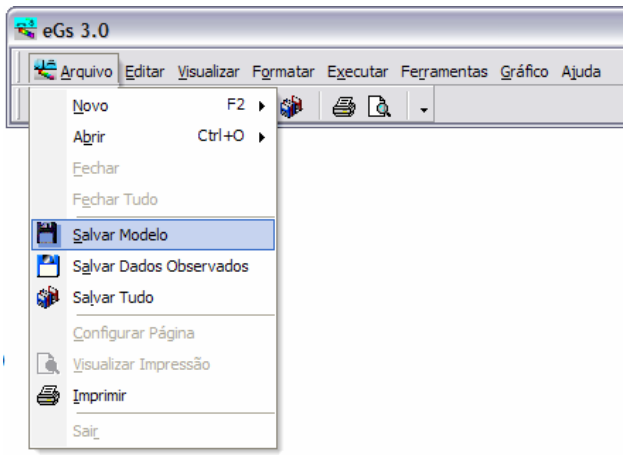


Figura A.14: Opção *Salvar Modelo* do Menu.

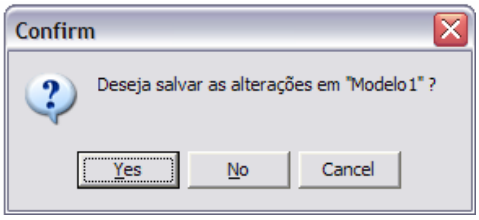


Figura A.15: Janela de aviso quando um *modelo geoelétrico* está sendo fechado, mas as alterações não foram salvas.

ii) A partir de um modelo geoelétrico salvo anteriormente

Para abrir uma janela "Modelo" com informações editadas anteriormente, primeiramente deve ser escolhido o tipo de problema na janela "Métodos", depois basta seguir um dos dois caminhos seguintes:

- Janela Seleção→ Modelo → Abrir (Figura A.16);
- Menu→Arquivo→Abrir→ Modelo (Figura A.17).

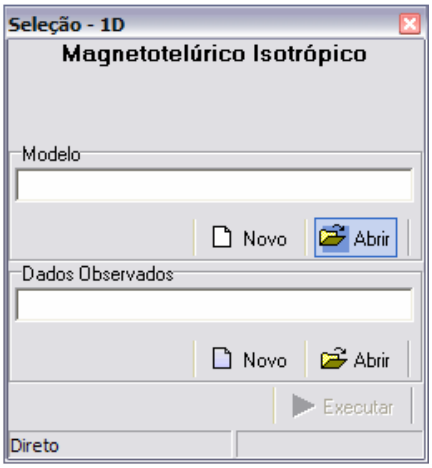


Figura A.16: Janela "Seleção" para abrir um *modelo geoelétrico* salvo.

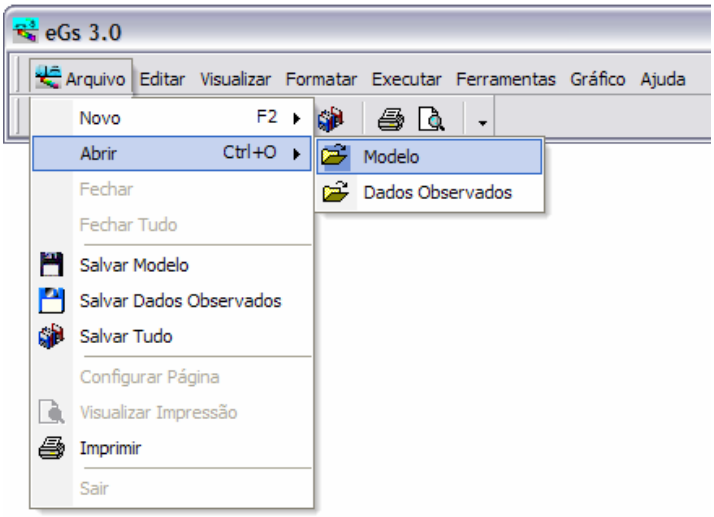


Figura A.17: Abrir um *modelo geoelétrico* salvo através do Menu.

Surgirá a janela "Abrir", mostrada na Figura A.18. Deve-se procurar arquivos com a extensão *.MGE. Em seguida a janela da Figura A.19 para reedição.

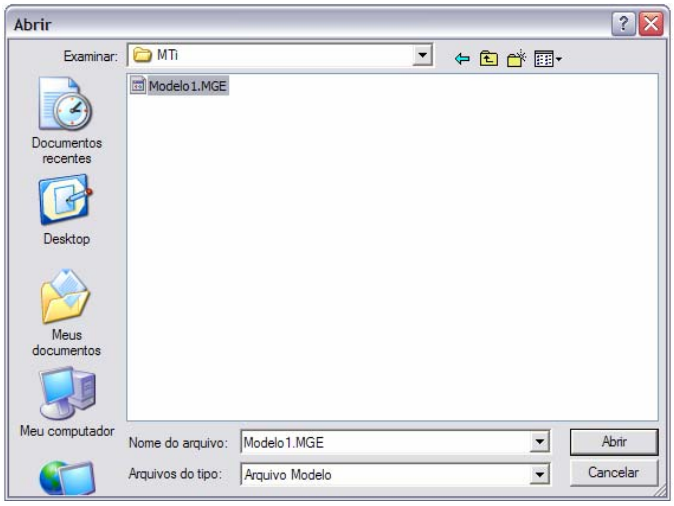


Figura A.18: Janela amigável para abertura de arquivos.

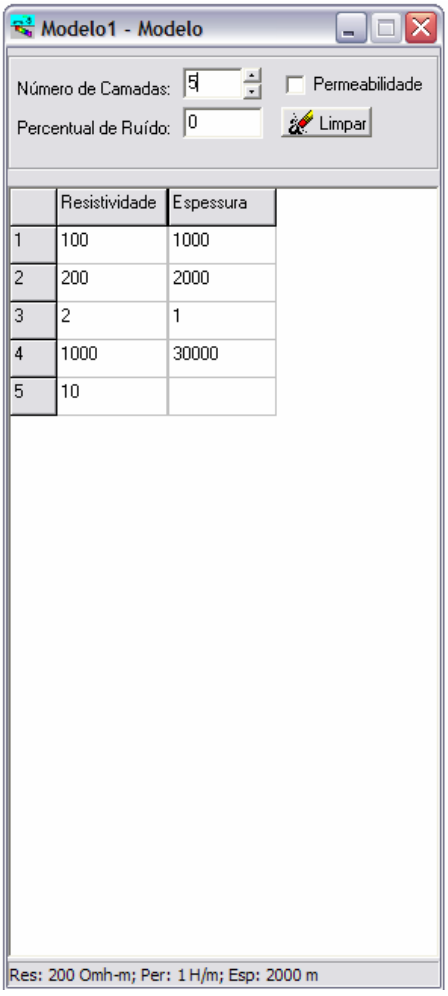


Figura A.19: Janela “Modelo” aberta para reedição de um arquivo salvo anteriormente.

3º) Construir os *dados observados*.

Depois de feita a escolha do tipo de problema, há duas maneiras de editar os dados observados:

i. A partir da edição de dados *observados* novos;
ii. A partir de *dados observados* salvo anteriormente.

i) A partir da edição de dados observados novos:

Para abrir uma janela "Dados Observados" para editar informações novas, primeiramente deve ser escolhido o tipo de problema na janela "Métodos", depois basta seguir um dos dois caminhos seguintes:

- Janela Seleção → Dados Observados → Novo (Figura A.20);
- Menu → Arquivo → Novo → Dados Observados (Figura A.21).

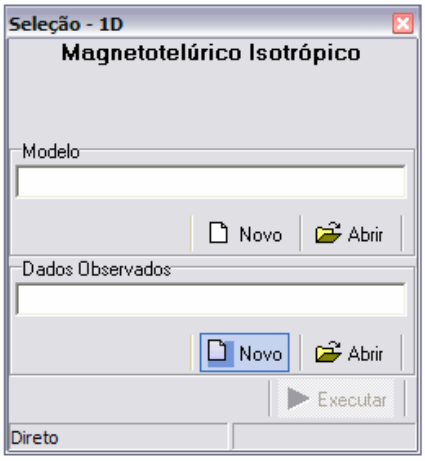


Figura A.20: Janela "Seleção" para abrir a janela "Dados Observados" nova.

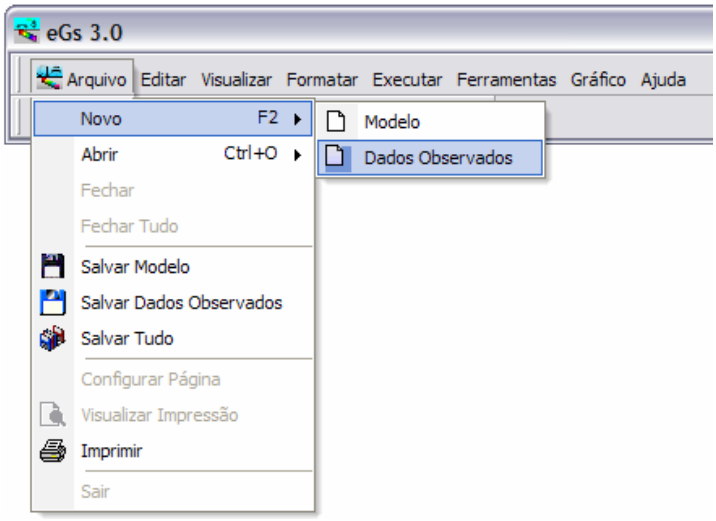


Figura A.21: Abrir uma janela “Dados Observados” nova através do Menu.

Surgirá a janela “Dados Observados” mostrada na Figura A.22.

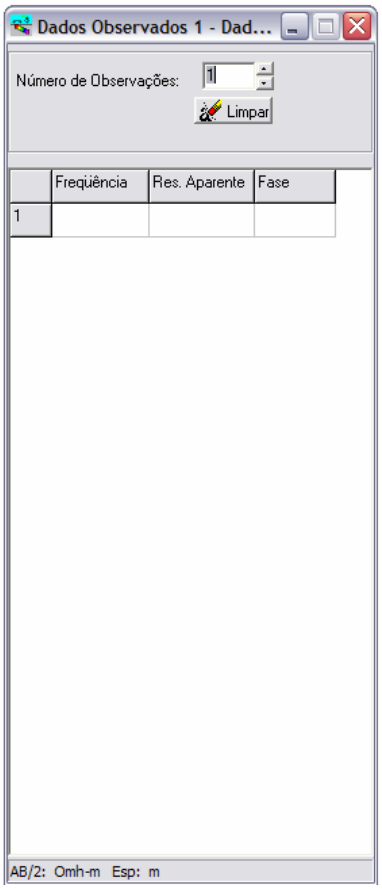


Figura A.22: Nova janela “Dados Observados” para início da edição.

Como exemplo, usou-se 26 observações de campo com as seguintes freqüências: 0.01, 0.01585, 0.02512, 0.03981, 0.06309, 0.1, 0.1585, 0.2512, 0.3981, 0.6309, 1, 1.585, 2.512, 3.981, 6.309, 10, 15.85, 25.12, 39.81, 63.09, 100, 158.5, 251.2, 398.1, 630.9 e 10000 Hz (Figura A.23).

No problema direto a resistividade aparente e fase não são exigidas somente se o objetivo for a interpretação dos dados por tentativa e erro usando a modelagem numérica, mas não há restrição caso o usuário não deseje editar essas duas propriedades físicas nesta etapa.

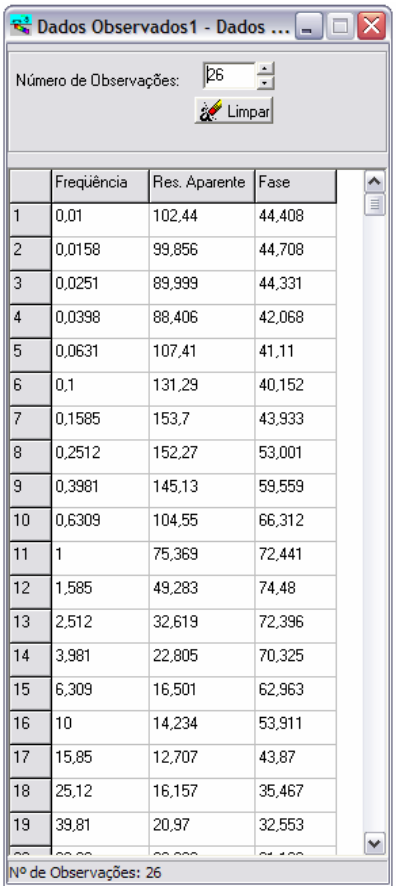


Figura A.23: Nova janela “Dados Observados” editada.

O item *Número de Observações* da janela “Dados observados” tem a função de alterar as linhas da tabela, as quais representam cada observação, que podem ser modificadas através do teclado digitando valores da freqüência, da resistividade aparente e fase, respectivamente, para a primeira, segunda e terceira colunas. O botão limpar, tem a finalidade de deixar em branco todos os campos da grade.

Para salvar os *dados observados* pode ser de duas maneiras: através do item salvar Dados Observados do menu (Figura A.24) ou quando o usuário desejar fechar a janela "Dados Observados" e não salvou as alterações (Figura A.25);

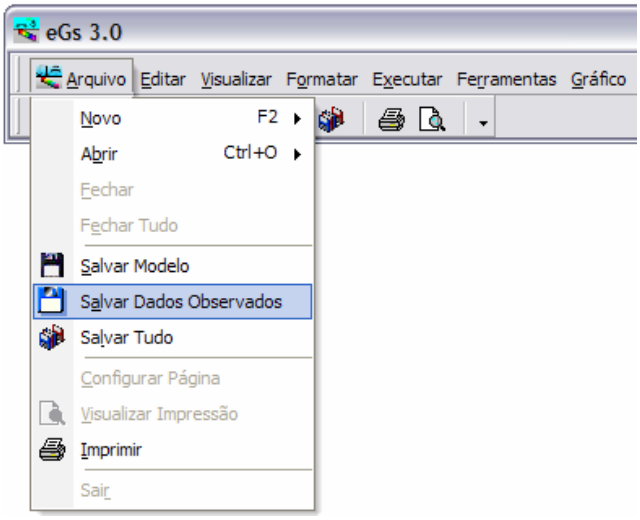


Figura A.24: Opção *Salvar Dados Observados* do Menu

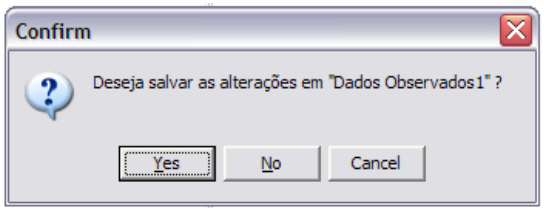


Figura A.25: Janela de aviso quando os *dados observados* estão sendo fechados, mas as alterações não foram salvas.

ii) A partir de dados observados salvos anteriormente:

Para abrir uma janela "Dados Observados" com informações editadas anteriormente, primeiramente deve ser escolhido o tipo de problema na janela "Métodos", depois basta seguir um dos dois caminhos:

- Janela Seleção→ Dados Observados → Abrir (Figura A.26);
- Menu→Arquivo→Abrir→ Dados Observados (Figura A.27).

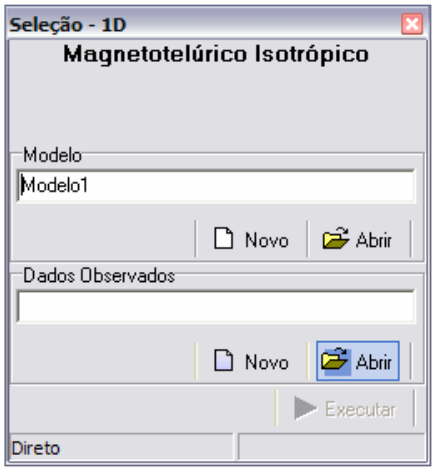


Figura A.26: Janela "Seleção" para abrir *dados observados* salvos.

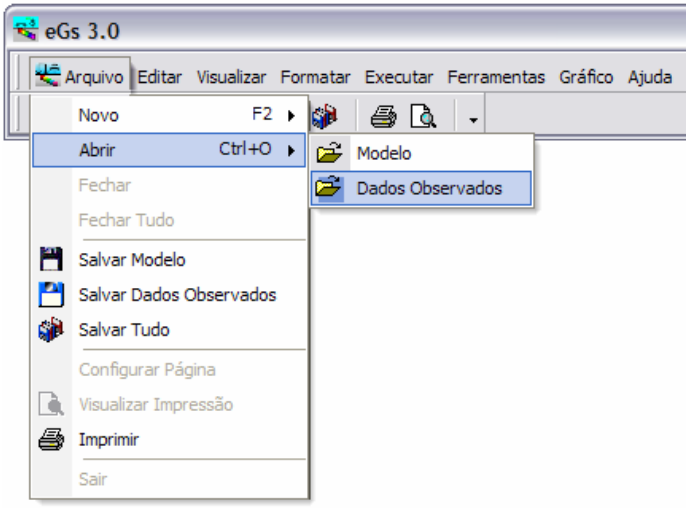


Figura A.27: Abrir os *dados observados* salvos através do Menu.

Surgirá a janela "Abrir", mostrada na Figura A.28. Deve-se procurar arquivos com a extensão *.OBS. Em seguida a janela da Figura A.29 para reedição.

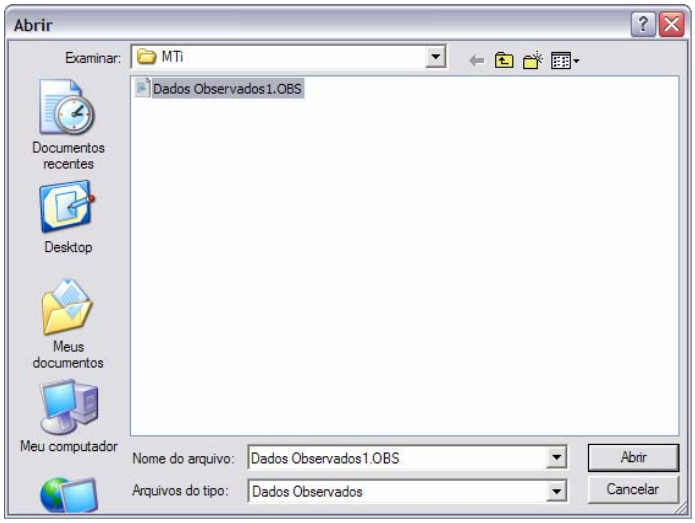


Figura A.28: Janela amigável para abertura de arquivos.

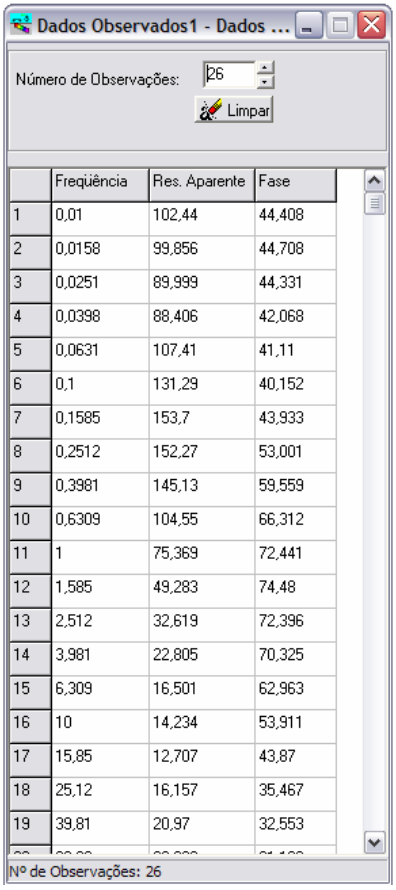


Figura A.29: Janela "Dados Observados" aberta para reedição de um arquivo salvo anteriormente.

4º) Executar a modelagem numérica:

Na eGs 3.0 a modelagem numérica só pode ser praticada após o *modelo geoelétrico* e *dados observados* estarem editados. Feito isso, o botão "Executar", que pode ser encontrado na janela "Seleção" (Figura A.30(a)) ou no item *Executar* do Menu (Figura A.30(b)), é ativado e com apenas um clique a modelagem é obtida, que pode ser com a geração de figuras para interpretação (Figura A.31) ou apenas armazena os dados de saída, isso vai depender da escolha feita pelo usuário nos subitens do Menu *Executar* (Figura A.30(b)).

(a) (b)

Figura A.30:(a) Janela "Seleção" com o botão *Executar* ativado (b) Itens *Executar Com Gráfico* e *Executar Sem Gráfico*.

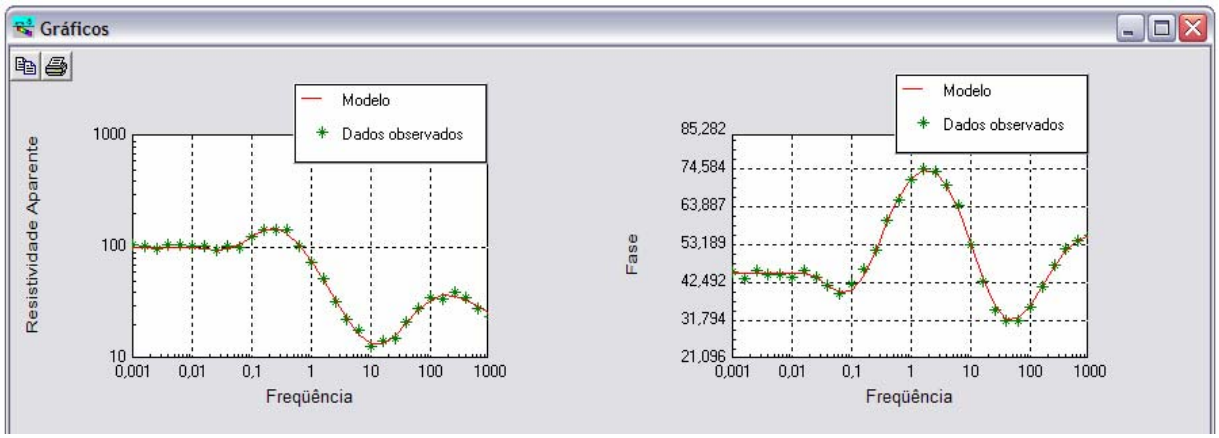


Figura A.31: Resultado da modelagem numérica com o item *Executar Com Gráfico* ativado.

Os arquivos do resultado do modelo *geoelétrico* e *dados observados* são salvos em uma pasta com o mesmo nome dado ao arquivo com extensão *.MGE. Por exemplo, se o *modelo geoelétrico* foi salvo em *C:\eGs\Exemplos\1D\MT\MTi\Mt1D.MGE*, então os dados de saída serão armazenados em uma pasta com o nome *Mt1D*. Veja Figura A.32.

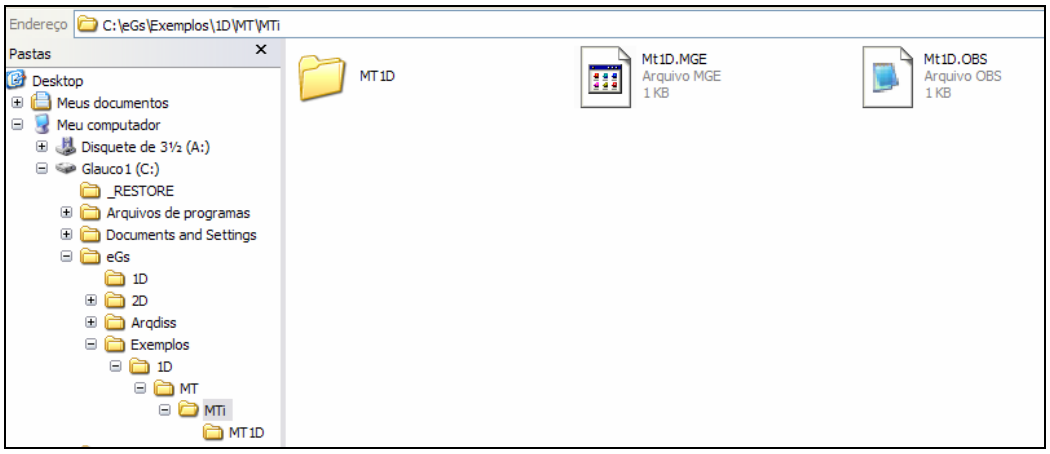


Figura A.32: Ilustração de uma modelagem usando o arquivo *Mt1D.MGE* como *modelo geoelétrico*.

Caso o usuário não queira salvar os arquivos *.MGE e *.OBS, os quais representam, respectivamente, o *modelo geolétrico* e os *dados observados*, mas deseja gerar o gráfico à interpretação, o aplicativo armazena temporariamente os arquivos de saída, constrói o gráfico e ao fechar a eGs 3.0 esses arquivos são apagados do disco rígido.

5º) Interpretação dos dados (gráficos de curvas)

Acionando-se a função *Executar com Gráfico*, os gráficos já aparecem automaticamente após a execução do programa. Mas caso o usuário prefira optar por executar a modelagem sem gráfico (botão *Executar sem Gráfico*) para interpretar os dados mais tarde, será necessário abrir os arquivos que contêm estes dados. Para abrir os arquivos que contêm os dados de resposta do modelo ou os dados observados, o usuário deve clicar em Menu→Gráfico→Curvas→ Abrir Arquivo (Figura A.33). A caixa de diálogo padrão do *Windows* aparece para que o usuário possa procurar o arquivo. No caso do método magnetotelúrico, o arquivo escolhido deve ser do tipo *.RAF ou *.OBS (com dados do método).

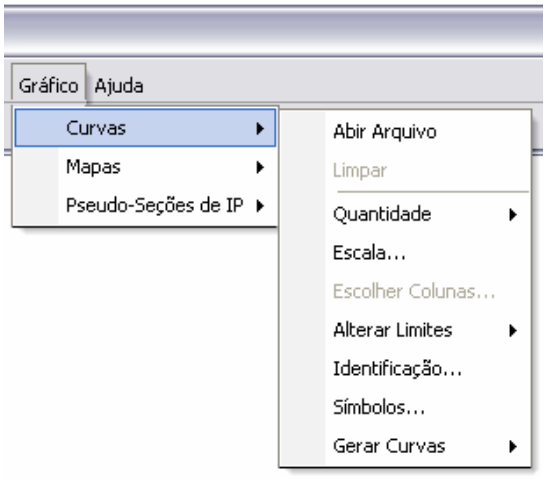
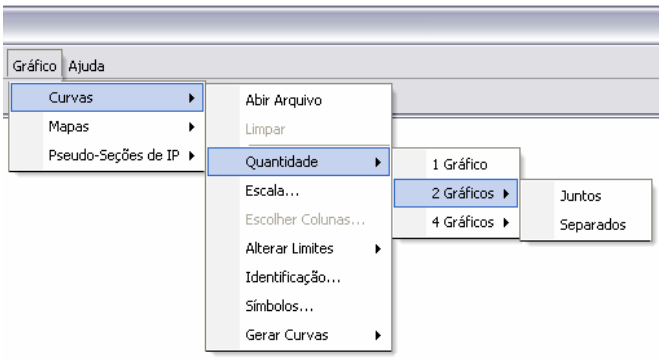
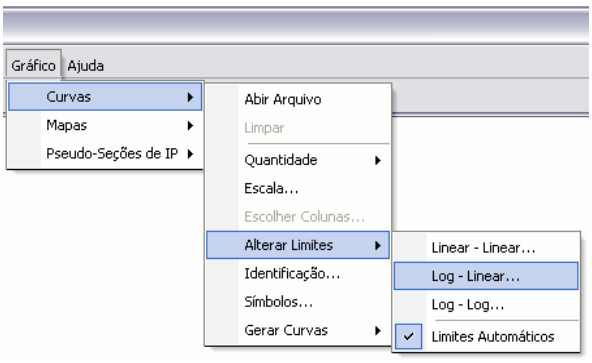


Figura A.33: Item *Abrir Arquivo*

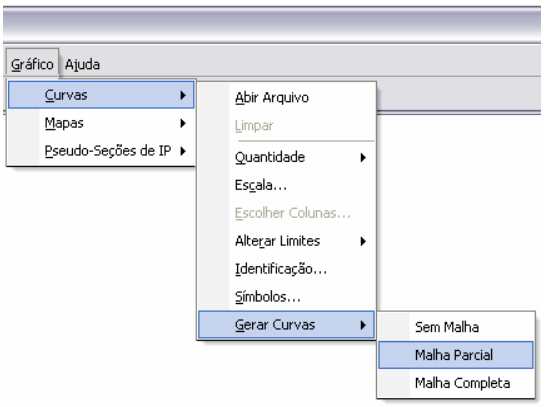
Caso o arquivo não tenha nenhum problema com os seus dados, já é possível fazer um gráfico. Basta escolher uma das opções no subitem *Gerar Curvas* (*Sem malha, Malha parcial, Malha completa*). A Figura A.34.(c) mostra as opções do subitem *Gerar Curvas*.



(a)



(b)



(c)

Figura A.34: Sub-menus do menu *Gráfico*. A figura (a) mostra os subitens do item *Quantidade* (um, dois ou quatro gráficos, juntos ou separados). A figura (b) exibe os subitens do item *Alterar Limites* que permitem alterar os intervalos de visualização dos gráficos nas escalas *Linear-Linear, Log-Linear ou Log-Log*, ou escolher a opção *Limites Automáticos*. A figura (c) mostra os subitens do item *Gerar Curvas*, os quais permitem escolher a malha (grade) que cobre os gráficos (*Sem malha, Malha parcial* ou *Malha completa*).

A eGs3.0 possui diversas opções para fazer os seus gráficos. Dentre estas opções temos:

- *Quantidade:* Permite escolher quantos gráficos vão aparecer de uma só vez (Figura A.34.(a)).
- *Escala* (Tipo de gráfico): Permite escolher a escala que será usada em cada um dos gráficos. Dentre as escalas usadas temos *linear-linear* (linear nos eixos X e Y), *log-linear* (linear no eixo X e logarítmica no eixo Y) e *log-log* (logarítmica nos eixos X e Y). A janela que permite fazer este tipo de alteração está ilustrada na Figura A.35.

Figura A.35: Janela que permite escolher as escalas de cada gráfico.

- *Escolher colunas:* Permite escolher quantas e quais colunas do arquivo serão usadas para cada gráfico (janela na Figura A.36).

Figura A.36: Janela que permite escolher quais colunas do arquivo serão usadas para gerar o gráfico.

- *Alterar limites* (Intervalos de visualização): Permite escolher os intervalos de visualização de cada gráfico (Figura A.37) de acordo com a escolha de um dos subitens de *Alterar limites: linear-linear, log-linear, log-log* (Figura A.34.(b)).

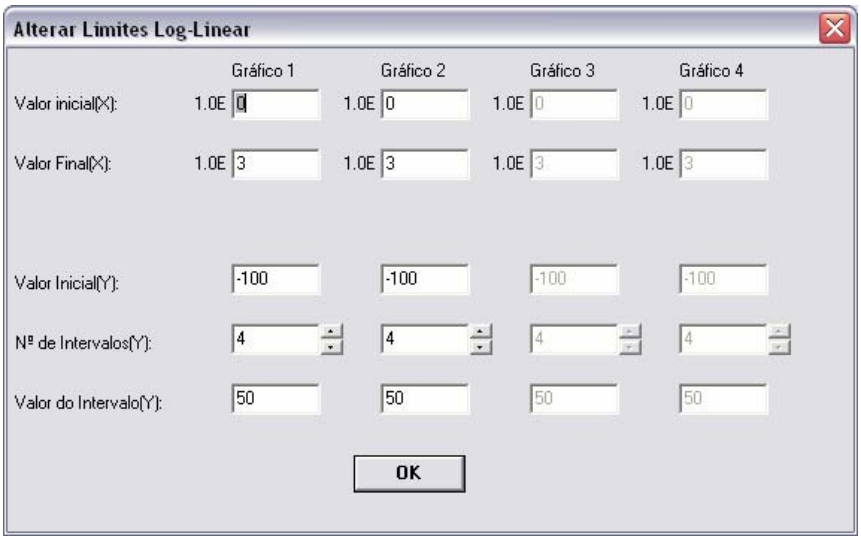


Figura A.37: Janela que permite escolher os intervalos de visualização dos gráficos. Esta janela em particular permite alterar os intervalos de visualização de gráficos em escala Log-Linear.

- *Identificação:* Permite escolher os textos que identificam a janela, os eixos e as curvas (Figura A.38).

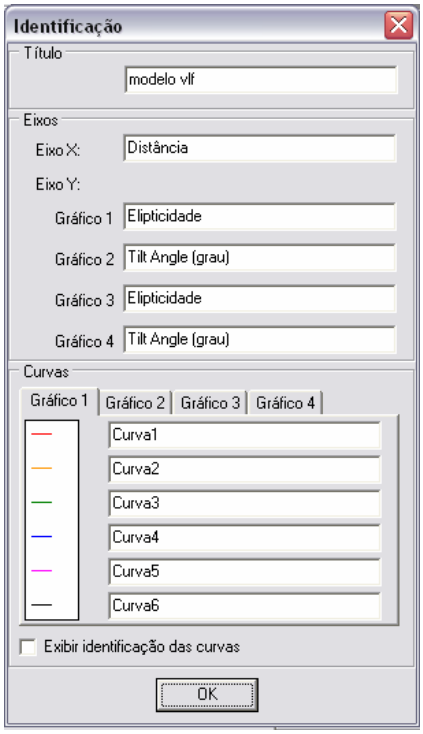


Figura A.38: Janela que permite escolher os textos que identificam a janela, os eixos e as curvas.

- *Símbolos:* Permite escolher quais símbolos vão representar os pontos no gráfico (Figura A.39).

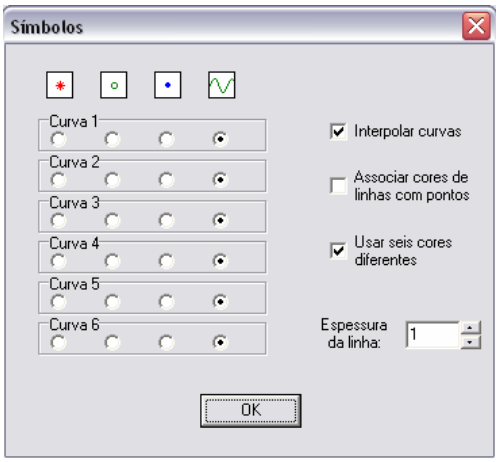


Figura A.39: Janela que permite escolher os símbolos que representarão os pontos nos gráficos.

- *Gerar Curvas* (Grade)*:* Permite escolher que tipo de malha (grade) vai cobrir os gráficos. Este item está subdividido em *Sem malha, Malha parcial* e *Malha completa* (Figura A.34.(c)).

Para comparar os dados armazenados em mais de um arquivo, o usuário deve primeiro gerar as colunas que deseja do arquivo que está aberto, usando a caixa de diálogo *Escolher colunas* (Figura A.36). Em seguida, sem limpar o gráfico que já está na tela, o usuário deve abrir o próximo arquivo, sendo que este novo arquivo deve ter a primeira coluna da sua matriz igual à do arquivo anterior, caso contrário não é possível fazer o gráfico. É importante que não se limpe o gráfico que está na tela antes de abrir o novo arquivo, caso contrário os dados do arquivo anterior são apagados da memória. Finalmente, caso não haja nenhum problema com o novo arquivo que foi aberto, a caixa de diálogo *Escolher colunas* aparece automaticamente para o usuário escolher quais colunas do novo arquivo serão usadas para gerar as curvas. Após clicar no botão *OK* os dados do novo arquivo gerarão os gráficos junto com os dados do(s) arquivo(s) anterior(es).

A.3.2 – 2D (duas dimensões)

A seguir, vamos mostrar um exemplo de modelagem 2D usando o método IP e Resistividade.

A.3.2.1 – IP e Resistividade 2D – Problema Direto

Para utilizar a eGs em uma aplicação geofísica que utiliza o método IP num modelo 2D, deve-se seguir estes passos:

1º) Escolher o tipo de problema

Para construir um modelo novo ou abrir um modelo do arquivo, primeiramente é necessário escolher, na janela "Métodos" (Figura A.40), a dimensão 2D, em seguida dar dois clicks em + *IP e Resistividade* e escolher o tipo de problema (neste caso problema direto).

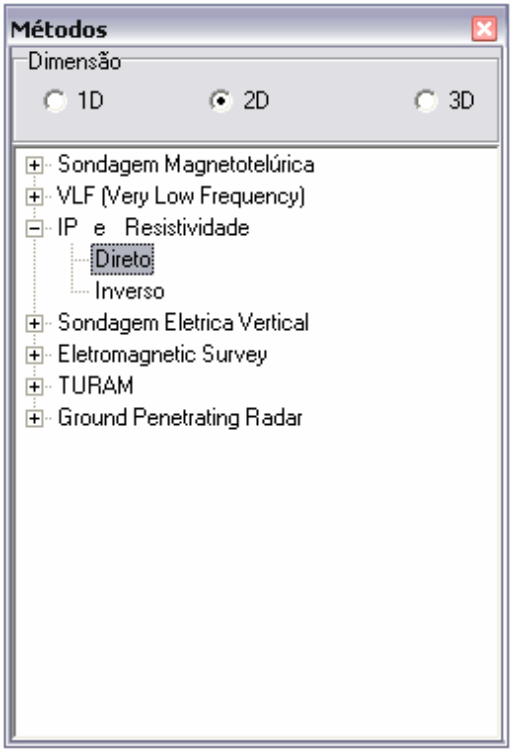


Figura A.40: Janela Métodos

2º) Chamar a janela usada para editar o modelo

Depois de feita a escolha do tipo de problema, há duas maneiras de editar um *modelo geoelétrico*:

- A partir de um *modelo geoelétrico* novo;
- A partir de um *modelo geoelétrico* salvo anteriormente.

i) A partir de um modelo geoelétrico novo:

Para abrir uma janela "Modelo" nova, primeiramente deve ser escolhido o tipo de problema na janela "Métodos", depois basta seguir um dos dois caminhos:

- Janela Seleção→ Modelo → Novo (Figura A.41);
- Menu→Arquivo→Novo→ Modelo (Figura A.42).

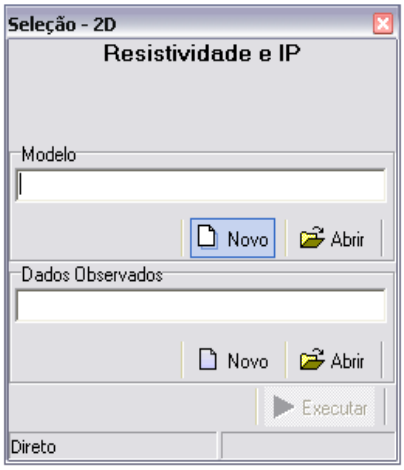


Figura A.41: Janela "Seleção" para abrir um *modelo geoelétrico* novo.

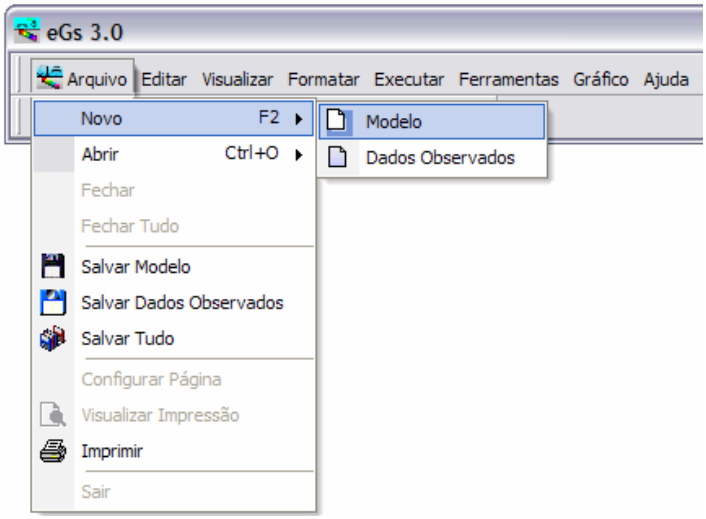


Figura A.42: Abrir um *modelo geoelétrico* novo através do Menu.

ii) A partir de um modelo geoelétrico salvo anteriormente

Para abrir uma janela "Modelo" com informações editadas anteriormente, primeiramente deve ser escolhido o tipo de problema na janela "Métodos", depois basta seguir um dos dois caminhos:

- Janela Seleção→ Modelo → Abrir (Figura A.43);
- Menu→Arquivo→Abrir→ Modelo (Figura A.44).

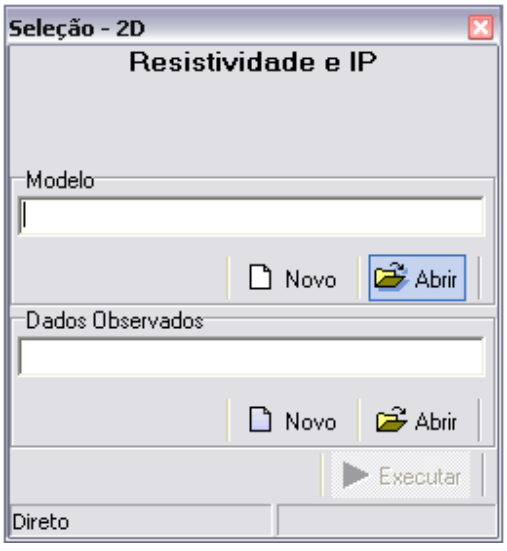


Figura A.43: Janela "Seleção" para abrir um *modelo geoelétrico* salvo.

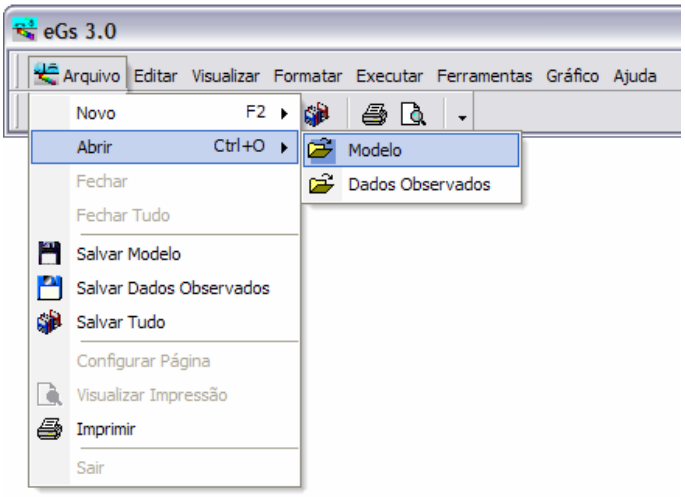


Figura A.44: Abrir um *modelo geoelétrico* salvo através do Menu.

Após chamar um modelo novo ou abrir um do arquivo, aparecerá a janela usada para edição do modelo. A Figura A.45 mostra esta janela.

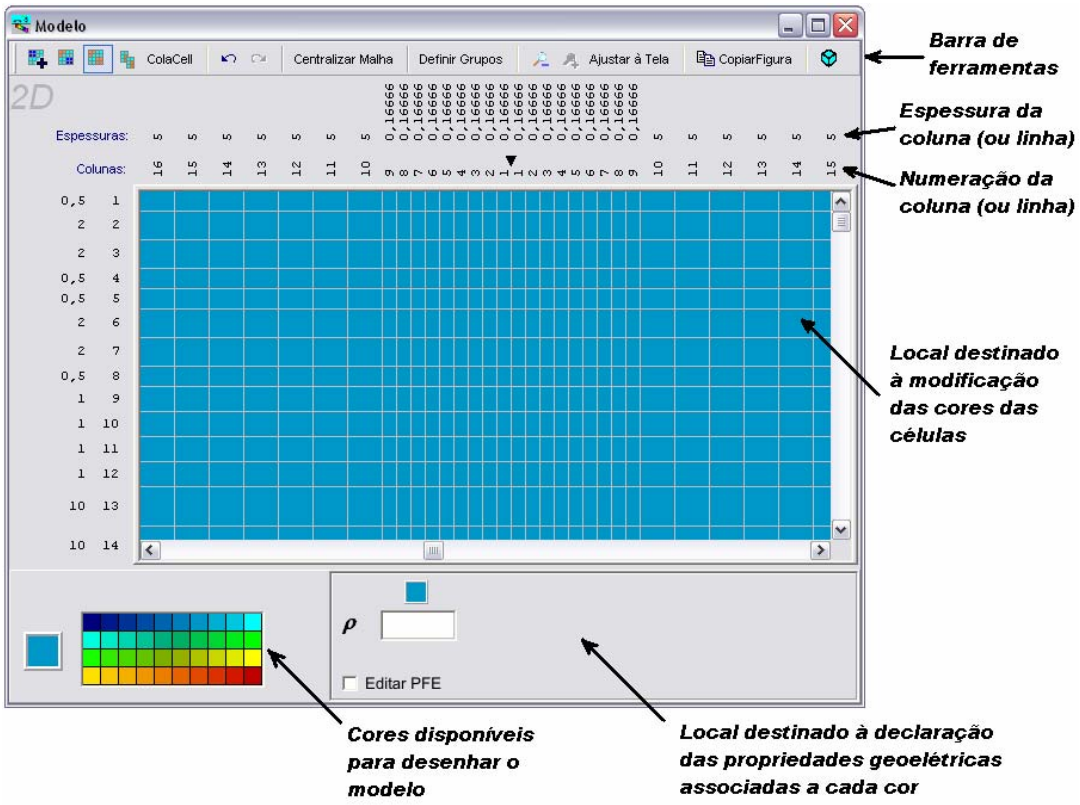


Figura A.45: Janela usada para editar o modelo 2D.

Na janela usada para desenhar o *modelo geoelétrico* (Figura A.45) estão todas as opções usadas para alterar a forma e as propriedades elétricas do modelo. A barra de ferramentas da janela usada para desenhar o *modelo geoelétrico* está ilustrada na Figura A.46. Abaixo temos uma breve explicação da função de cada item da barra.

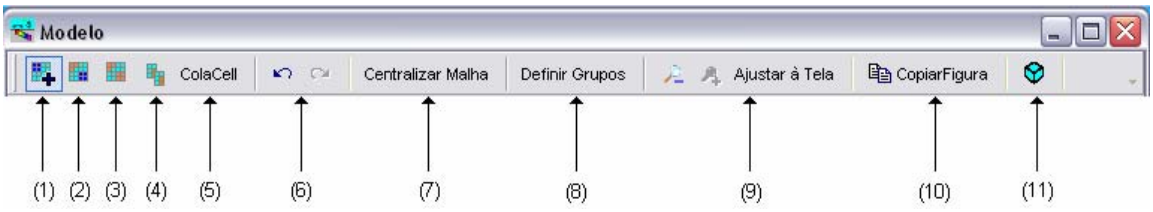


Figura A.46: Barra de ferramentas da janela usada para desenhar o *modelo geoelétrico*.

(1): Permite criar um corpo retangular (arrastando-se o cursor do mouse na região onde as células terão suas cores modificadas);

(2): Permite mudar a cor de um corpo (clicando-se o cursor do mouse em uma das células do corpo);

(3): Permite mudar a cor das células de mesma cor (clicando-se o cursor do mouse em uma das células da cor que deve ser modificada);

(4): Permite copiar as cores das células de uma região retangular (arrastando-se o cursor do mouse na região onde as células terão suas cores copiadas);

(5): Permite colar as células da região copiada (clicando-se o cursor do mouse no local onde a região copiada será colada);

(6): Desfaz ou refaz a última ação;

(7): Move a barra de rolagem para centralizar a malha;

(8): Chama a janela que permite definir as dimensões da malha;

(9): Ferramentas de zoom;

(10): Copia para a área de transferência o trecho da malha que está visível;

(11): Chama a janela que permite visualizar o modelo em 3D.

3º) Editar o modelo

Na fase de criação ou de modificação de um modelo, o usuário pode alterar as características do mesmo de três formas. A seqüência para se fazer esta modificação deve ser, preferencialmente, a apresentada a seguir. As características do modelo que podem ser alteradas são:

i. Dimensões (células da malha);
ii. Forma;
iii. Propriedades físicas.

i) Definir as dimensões do modelo

Para definir as dimensões da malha, o usuário deve clicar no botão *Definir Grupos* (item (8) da Figura A.46) na barra de ferramentas da janela usada para editar o modelo. A janela ilustrada na Figura A.47 aparece para que o usuário possa declarar as dimensões da malha. Nesta janela o usuário declara o número de nós da malha e as espessuras das células da mesma. As espessuras das células são declaradas em grupos. Em cada direção, defini-se o número de grupos de células, o número de células em cada grupo e a espessura das células do grupo.

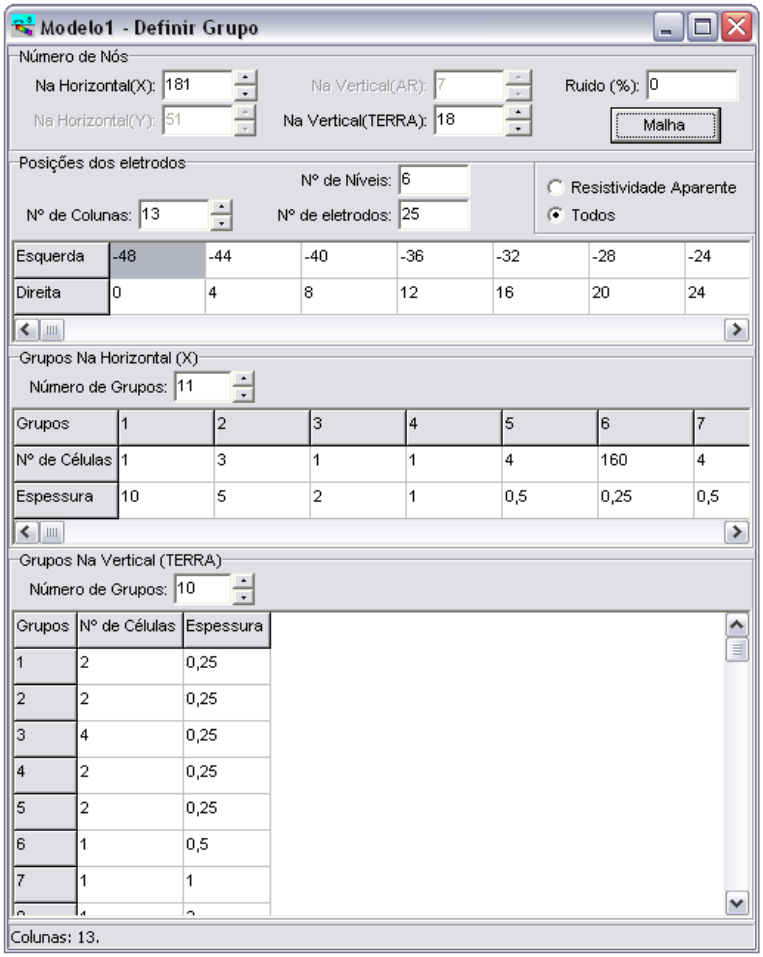

Modelo1 - Definir Grupo
Número de Nós
Na Horizontal(X): 181
Na Horizontal(Y): 51
Na Vertical(AR): 7
Na Vertical(TERRA): 18
Ruido (%): 0
Malha
Posições dos eletrodos
Nº de Níveis: 6
Nº de Colunas: 13
Nº de eletrodos: 25
Resistividade Aparente
Todos
Esquerda -48 -44 -40 -36 -32 -28 -24
Direita 0 4 8 12 16 20 24
Grupos Na Horizontal (X)
Número de Grupos: 11
Grupos 1 2 3 4 5 6 7
Nº de Células 1 3 1 1 4 160 4
Espessura 10 5 2 1 0,5 0,25 0,5
Grupos Na Vertical (TERRA)
Número de Grupos: 10
Grupos Nº de Células Espessura
1 2 0,25
2 2 0,25
3 4 0,25
4 2 0,25
5 2 0,25
6 1 0,5
7 1 1
Colunas: 13.

Figura A.47: Janela usada para definir as dimensões da malha usada no *modelo geoelétrico* do método IP e Resistividade 2D.

ii) Definir a forma do modelo

Para definir a forma do modelo, o usuário deve associar cada tipo de material (rocha) que compõe o modelo a uma cor. Para desenhar os corpos e camadas no modelo, o usuário pode simplesmente selecionar uma nova cor e clicar nas devidas células, ou usar as ferramentas para a edição de modelos (itens (1), (2), (3), (4) e (5) da Figura A.46). A Figura A.48 mostra o modelo construído.

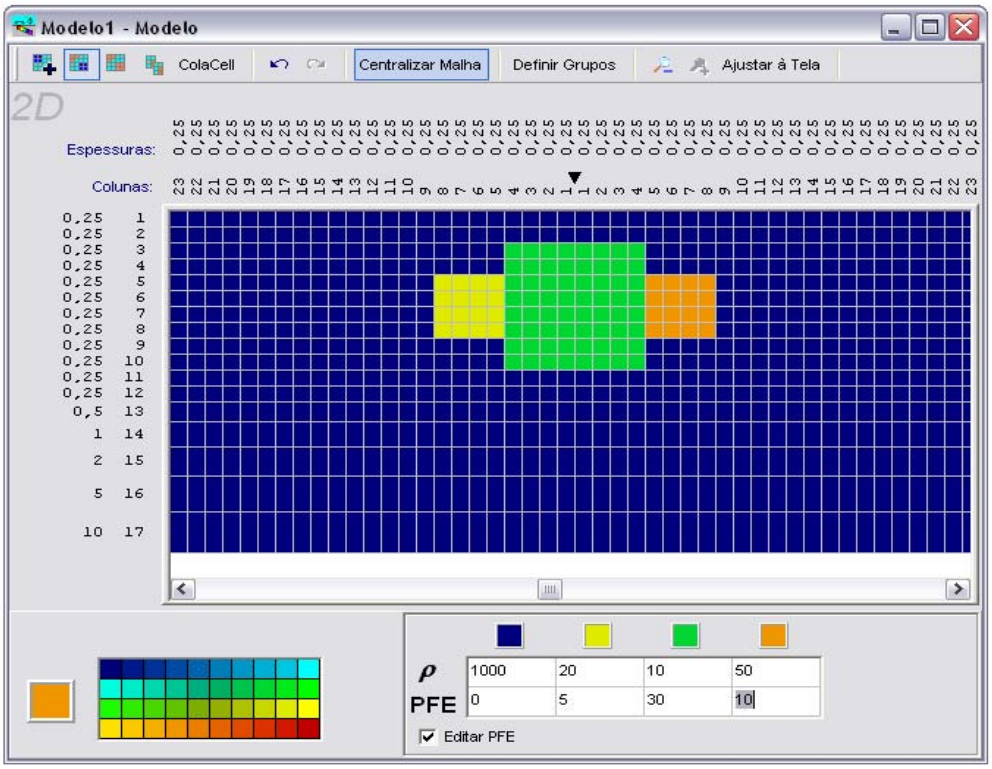


Figura A.48: Ilustração da forma do *modelo geoelétrico* usado na modelagem usando o método IP e Resistividade 2D.

iii) Declarar as propriedades físicas do modelo

Para associar os valores de propriedades físicas (que para o caso do IP são a resistividade e o PFE) a cada cor, o usuário deve preencher os campos que aparecem automaticamente no canto direito-inferior da janela (veja a Figura A.45). A Figura A.49 mostra com maior destaque estes campos.

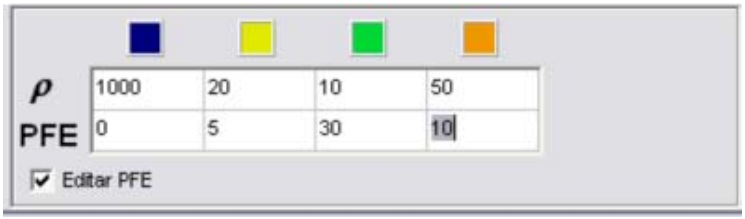


Figura A.49: Campos destinados à declaração das propriedades físicas do *modelo geoelétrico*.

4º) Salvar o modelo

O *modelo geoelétrico* pode ser salvo de duas maneiras: através do item *salvar modelo* do menu (Figura A.50) ou quando o usuário desejar fechar a janela "Modelo" e não salvou as alterações (Figura A.51);

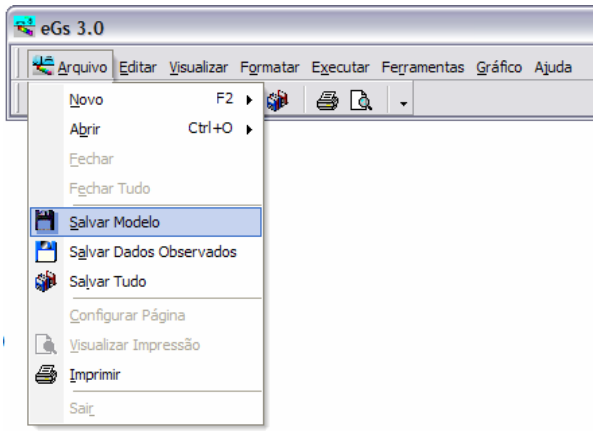


Figura A.50: Opção *Salvar Modelo* do Menu

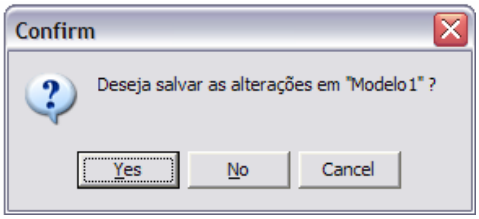


Figura A.51: Janela de aviso quando um *modelo geoelétrico* está sendo fechado, mas as alterações não foram salvas.

5º) Construir os *dados observados*.

Depois de feita a escolha do tipo de problema, há duas maneiras de editar os dados observados:

i. A partir de dados observados novos;
ii. A partir de dados observados salvo anteriormente.

i) A partir de dados observados novos:

Para abrir uma janela “Dados Observados” nova, primeiramente deve ser escolhido o tipo de problema na janela “Métodos”, depois basta seguir um dos dois caminhos:

- Janela Seleção→ Dados Observados → Novo (Figura A.52);
- Menu→Arquivo→Novo→ Dados Observados (Figura A.53).

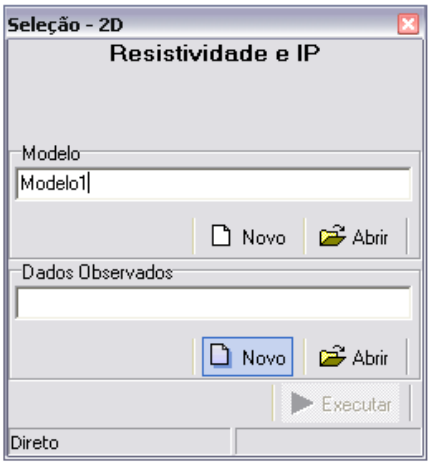


Figura A.52: Janela “Seleção” para abrir a janela “Dados Observados” nova.

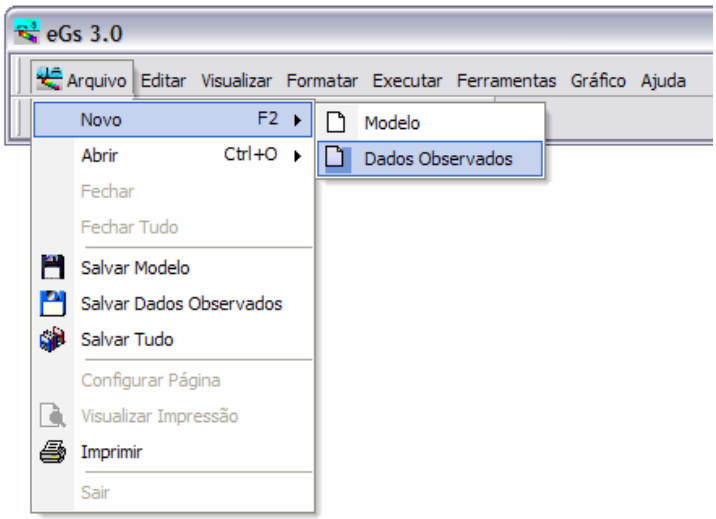


Figura A.53: Abrir uma janela “Dados Observados” nova através do Menu.

Surgirá a janela “Dados Observados” mostrada na Figura A.54.

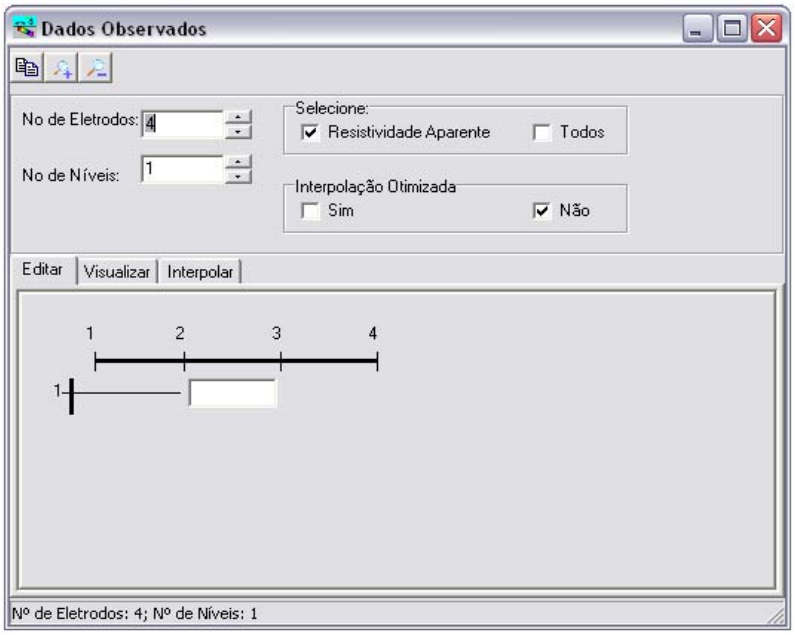


Figura A.54: Nova janela "Dados Observados" para início da edição.

A janela da Figura A.55 mostra a edição das observações de campo com 5 níveis e 16 eletrodos.

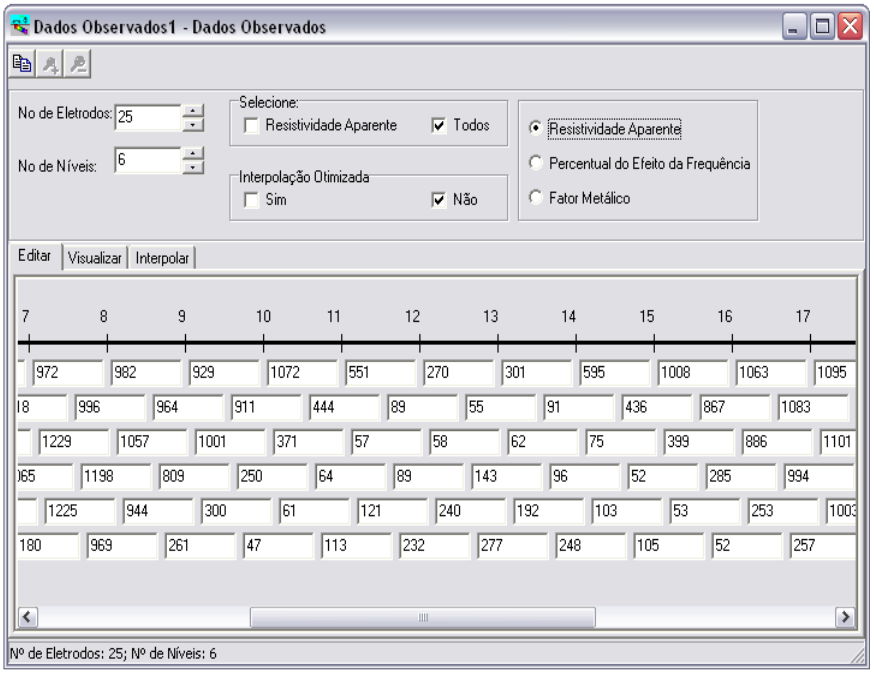


Figura A.55: Nova janela "Dados Observados" editada.

Os números de eletrodos e de níveis são alterados nas caixas de texto *No de Eletrodos* e *No de Níveis*, os quais devem obedecer à seguinte condição: No de Eletrodos ≥ No de Níveis +3. Caso o usuário não respeite a condição, o programa corrige acrescentando (ou diminuindo) eletrodos (ou níveis), de modo que sempre a condição seja satisfeita.

Caso a opção *Selecione* esteja marcada em *todos* (Figura A.56), indica que o usuário deseja editar ou visualizar resistividade aparente, PFE e fator metálico, aparecendo os itens mostrados na Figura A.57 para escolher qual a propriedade o usuário quer editar.

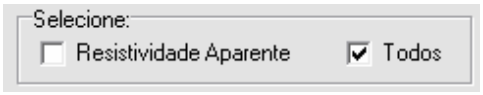


Figura A.56: Escolha para edição da resistividade aparente, percentual do efeito da freqüência e fator metálico ou apenas da resistividade aparente.

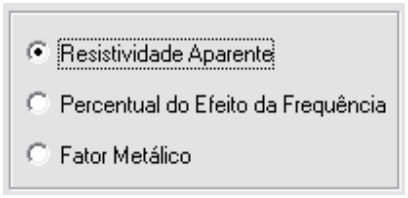


Figura A.57: Escolha do tipo de propriedade que o usuário deseja editar.

Os *dados observados* podem ser salvos de duas maneiras: através do item *Salvar Dados Observados* do menu (Figura A.58) ou quando o usuário desejar fechar a janela "Dados Observados" e não salvou as alterações (Figura A.59);

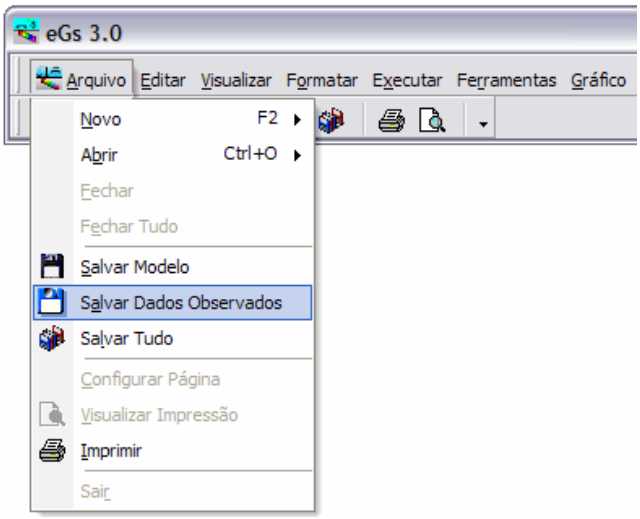


Figura A.58: Opção *Salvar Dados Observados* do Menu

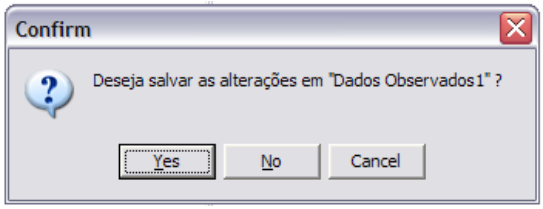


Figura A.59: Janela de aviso quando os *dados observados* estão sendo fechados, mas as alterações não foram salvas.

ii) A partir de dados observados salvos anteriormente

Para abrir uma janela "Dados Observados" com informações editadas anteriormente, primeiramente deve ser escolhido o tipo de problema na janela "Métodos", depois basta seguir um dos dois caminhos:

- Janela Seleção→ Dados Observados → Abrir (Figura A.60);
- Menu→Arquivo→Abrir→ Dados Observados (Figura A.61).

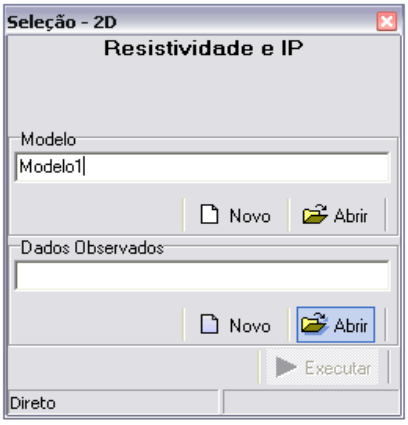


Figura A.60: Janela "Seleção" para abrir *dados observados* salvos.

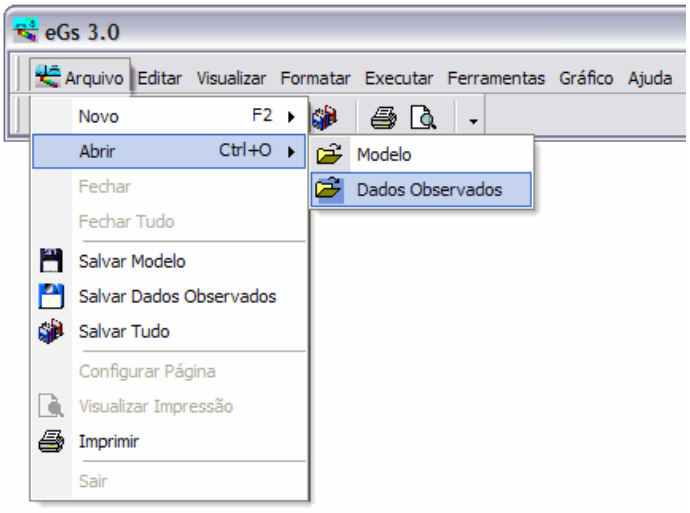


Figura A.61: Abrir os *dados observados* salvos através do Menu.

Surgirá a janela "Abrir Dados Observados", mostrada na Figura A.62. Deve-se procurar arquivos com a extensão *.OBS. Em seguida aparece a janela para reedição, mostrada na Figura A.63.

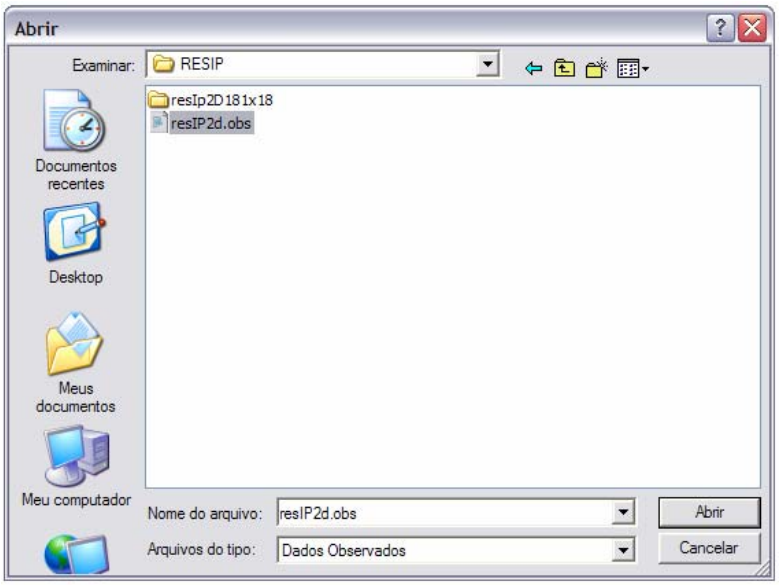


Figura A.62: Janela para abertura de arquivos.

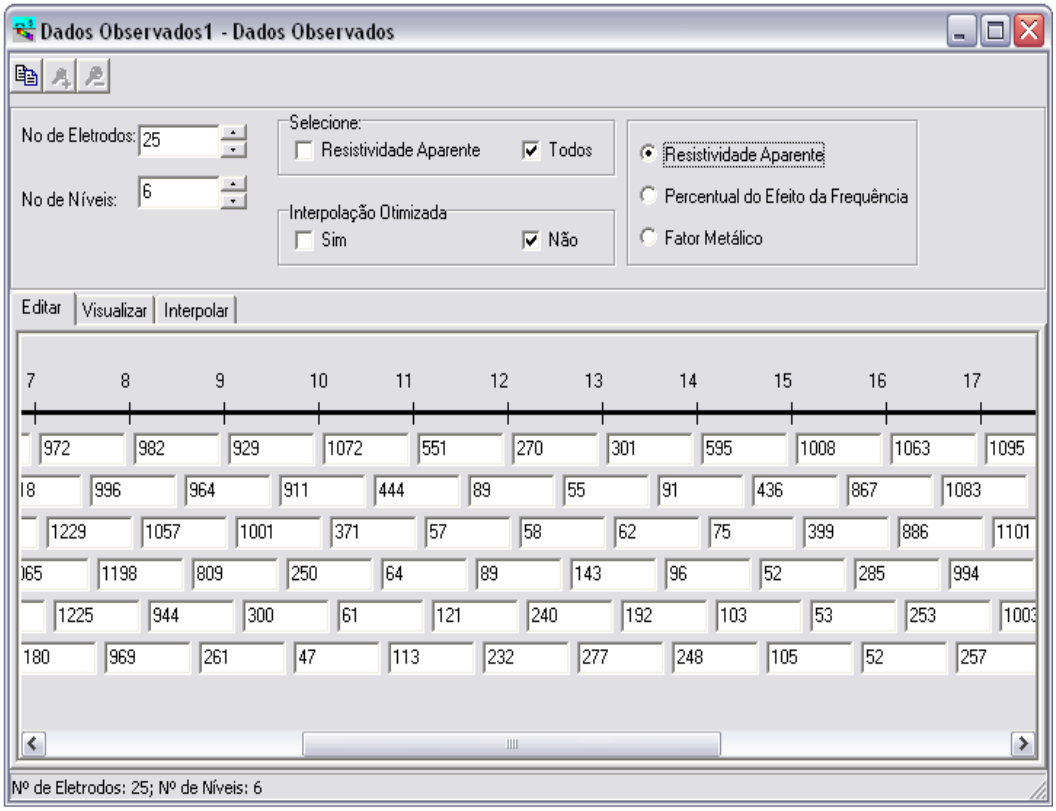
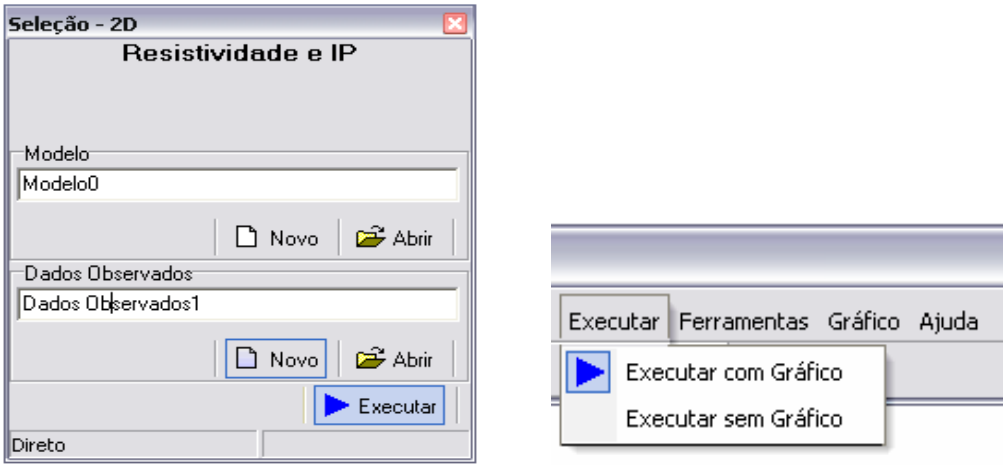


Figura A.63: Janela "Dados Observados" aberta para reedição de um arquivo salvo anteriormente.

6º) Executar a modelagem numérica:

Na eGs 3.0 a modelagem numérica só pode ser executada após o *modelo geoelétrico* e os *dados observados* estarem editados. Feito isso, o botão "Executar", que pode ser encontrado na janela "Seleção" (Figura A.64(a)) ou no item *Executar* do Menu (Figura A.64(b)), pode ser ativado e com apenas um clique a modelagem é obtida, que pode ser com a geração de figuras para interpretação (Figura A.65) ou apenas com o armazenamento dos dados de saída, dependendo da escolha feita pelo usuário nos subitens do Menu *Executar* (Figura A.64(b)).



(a) (b)

Figura A.64:(a) Janela "Seleção" com o botão *Executar* ativado. (b) Itens *Executar Com Gráfico* e *Executar Sem Gráfico*.

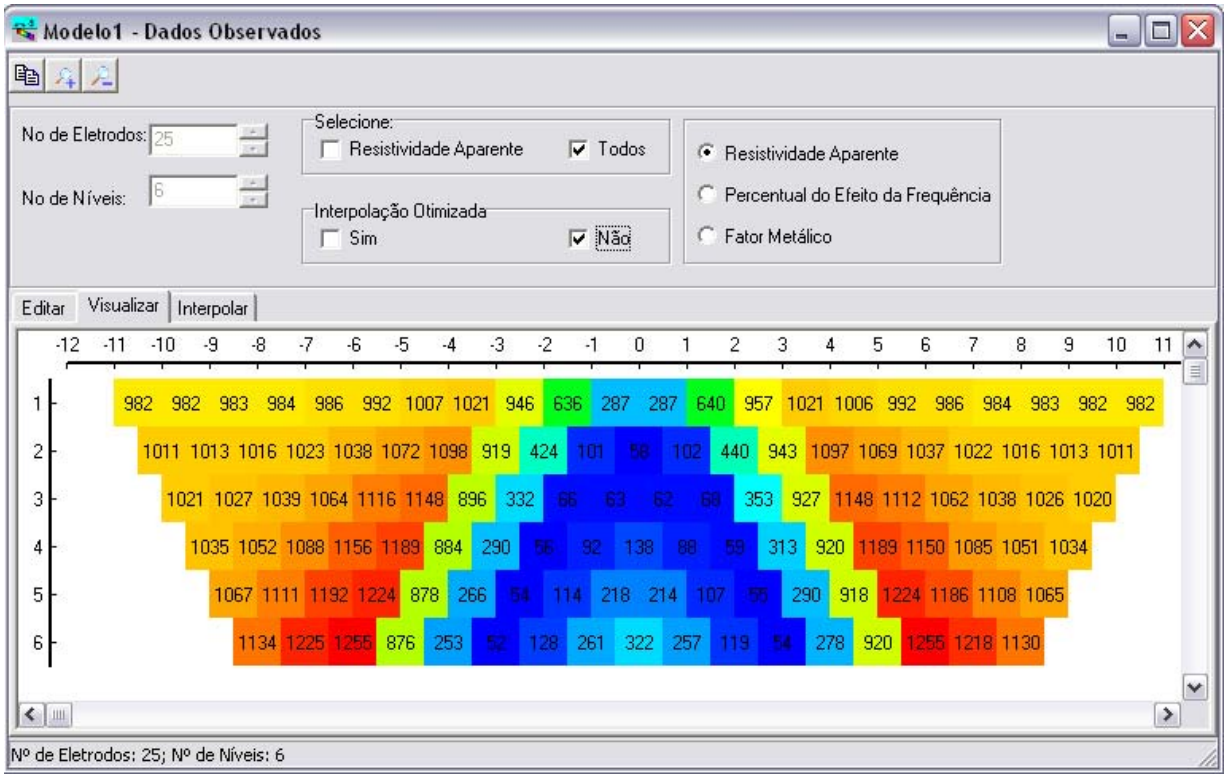


Figura A.65: Resultado da modelagem numérica com o item *Executar Com Gráfico* ativado.

Os arquivos do resultado do modelo *geoelétrico* e *dados observados* são salvos em uma pasta com o mesmo nome dado ao arquivo com extensão *.MGE. Por exemplo, se o *modelo geoelétrico* foi salvo em c:\eGs\Modelos\2D\ResIP\Modelo1.MGE, então os dados de saída serão armazenados em uma pasta com o nome Modelo1. Caso o usuário não queira salvar os arquivos *.MGE e *.OBS, os quais representam, respectivamente, o *modelo geolétrico* e os *dados observados*, o aplicativo armazena temporariamente os arquivos de saída e ao fechar a eGs 3.0 esses arquivos são apagados do sistema.

7º) Interpretação dos dados (pseudo-seções de IP e resistividade)

Os dados de resposta do modelo são visualizados na mesma janela usada para edição e visualização dos dados observados. A Figura A.66 mostra estes dados.

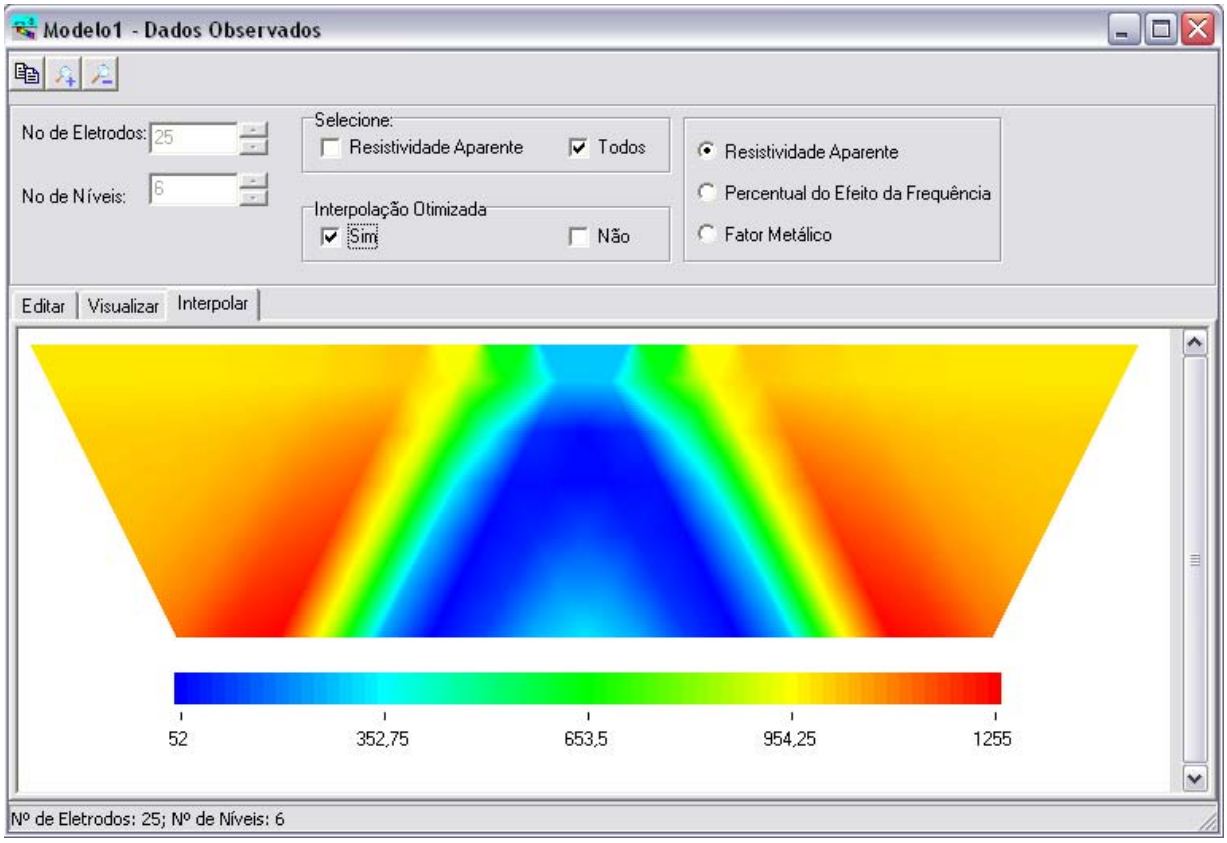


Figura A.66: Resposta do *modelo geoelétrico* do método IP e resistividade.

Em *Visualizar* e *Interpolar*, o usuário tem as seguintes opções de manipulação das figuras que representam as pseudo-seções:

→ Diminui as pseudo-seções nos tópicos *Visualizar* da janela;

→ Aumenta as pseudo-seções nos tópicos *Visualizar* da janela;

→ Copia as pseudo-seções nos tópicos *Visualizar* ou *Interpolar* da janela para a área de transferência.

A seção *Interpolar* tem a finalidade de mostrar ao usuário uma visualização mais suave do contraste das propriedades, como mostra a Figura A.66. Caso o usuário queira ver uma figura com melhor resolução das propriedades, basta escolher “sim” na opção *Interpolação Otimizada* (Figura A.67).

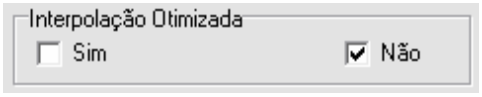


Figura A.67: Opção para interpolação das pseudo-seções de Resistividade e IP (deve estar marcada a opção “sim”)